University of Alberta

Evaluation, Design and Implementation of Multilevel DRAM

by

Gershom Birk

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment of the requirements for the degree of Master of Science.

Department of Electrical and Computer Engineering

Edmonton, Alberta

Fall 1999

National Library of Canada
Acquisitions and Bibliographic Services
395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale du Canada
Acquisitions et services bibliographiques
395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-47008-3

Canada

University of Alberta

Library Release Form

Name of Author: Gershom Birk

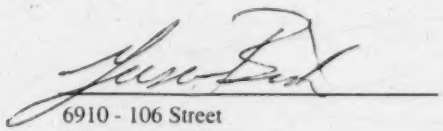
Title of Thesis: Evaluation, Design and Implementation of Multilevel DRAM

Degree: Master of Science

Year this Degree Granted: 1999

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except as hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.



6910 - 106 Street
Edmonton, Alberta
Canada T6H 2W3

Date: Aug. 27/99

University of Alberta

Faculty of Graduate Studies and Research

The undersigned certify that they have read, and recommended to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled Evaluation, Design and Implementation of Multilevel DRAM submitted by Gershom Birk in partial fulfillment of the requirements for the degree of Master of Science.

D S Elliott

Duncan Elliott, Co-Supervisor

Bruce Cockburn, Co-Supervisor

Martin Margala

Hong Zhang

Aug 27/99

To Trista.

ABSTRACT

Multilevel DRAM (MLDRAM) attempts to increase storage density by storing more than one bit per memory cell. Several different two-bit-per-cell schemes have been described in the literature and some have been implemented on large test chips. However, none of the schemes are currently used in production parts because they lack the robustness required to overcome the reduced noise margins characteristic of MLDRAM. The goal of this thesis is to explore the design, implementation and characteristics of MLDRAM to find a robust and economically manufacturable MLDRAM design. The first silicon implementation and characterization of P. Gillingham's MLDRAM is described. This thesis also proposes a new MLDRAM scheme that offers greater robustness and high speed at the expense of increased area overhead, with respect to the alternative MLDRAMs. Three past MLDRAM schemes and the new scheme are compared under consistent conditions in a simulation study.

ACKNOWLEDGMENTS

This research was supported by Micronet R&D, MOSAID Technologies Inc., the Canadian Microelectronics Corporation, the Natural Sciences and Engineering Research Council of Canada and the University of Alberta. I thank Valerie Lines, Randy Torrance, Roger Colbeck, Dan Clein and Peter Gillingham of MOSAID Technologies Incorporated for their technical support of MLDRAM research at the University of Alberta. I would also like to thank Albert Kwong for his help in the test chip design, Erika Peters for testing the test chip, and Ralph Nevins for help in keeping our CAD tools and workstations running smoothly. I would like to give special thanks to my supervisors Duncan Elliott and Bruce Cockburn for all they have taught me, and for providing me with the opportunity (and the desire) to study memories.

TABLE OF CONTENTS

Chapter 1: Introduction **1**

1.1 Semiconductor Memory 1

1.2 DRAM Technology 2

1.3 MLDRAM Technology 4

1.4 Thesis Outline 8

Chapter 2: DRAM and MLDRAM **10**

2.1 DRAM Manufacturing Processes 10

2.2 Basic DRAM Circuits and Design Techniques 11

2.2.1. The DRAM Array 11

2.2.2. The DRAM Core 15

2.2.3. The DRAM Periphery 19

2.3 DRAM Integrated Circuits 19

2.4 DRAM Operation 20

2.5 Embedded DRAMs 22

2.5.1. The HDRAM Macrocell 22

2.6 MLDRAM Methods and Circuits 23

2.6.1. Aoki's MLDRAM 24

2.6.2. Furuyama's MLDRAM 25

2.6.3. Gillingham's MLDRAM 28

2.6.4. Okuda's MLDRAM 32

2.7 MLDRAM Challenges 34

Chapter 3: Design of an MLDRAM Test Chip 36

3.1 Design Overview 36

3.2 Design Details 36

3.2.1. The DRAM Core 36

3.2.1.1 Sense Amplifiers 38

3.2.1.2 Bitline Twists and Dummy Cells 41

3.2.1.3 Wordline Drivers 43

3.2.1.4 Row and Column Address Scrambling 44

3.2.2. The Peripheral Circuitry 45

3.2.2.1 Row and Column Pre-decoders 45

3.2.2.2 Data I/O 45

3.2.2.3 Chip Pads 46

Chapter 4: Test Results 48

4.1 Test Equipment 48

4.2 Test of the ASIC DRAM Chip 48

4.2.1. DUT Board Design 49

4.2.2. Test Vector Generation 49

4.2.3. Access Time Tests 50

4.3 Test of the MLDRAM Chip 51

4.4 DUT Board Design 52

4.4.1. Testing 52

4.4.2. Results 53

4.4.3. Conclusions 54

Chapter 5: Description of an Improved MLDRAM 55

5.1 Detailed Description of the New MLDRAM Scheme 57

5.1.1. MLDRAM Circuitry 57

5.1.2. MLDRAM Operation 59

5.2 Discussion 65

Chapter 6: Simulation Study of Four MLDRAM Circuits 67

6.1 Experimental Setup 67

6.1.1. Controlled Parameters 67

6.1.2. MLDRAM Schematics 69

6.1.2.1 Furuyama's MLDRAM 70

6.1.2.2 Gillingham's MLDRAM 73

6.1.2.3 Okuda's MLDRAM 74

6.1.2.4 The New MLDRAM 76

6.1.3. Simulation Methodology 76

6.2 Results 78

6.3 Discussion 82

6.4 Conclusions 84

Chapter 7: Conclusions 85

7.1 MLDRAM Chip Designs 85

7.2 MLDRAM Test Chip Characterization 86

7.3 An Improved MLDRAM Scheme 86

7.4 MLDRAM Simulation Study 87

7.5 Future Work 87

Appendix A: Measuring Sensing Thresholds Using HSPICE 90

A.1 A Perl Script for Invoking HSPICE 90

A.2 The HSPICE Netlist Files Used 96

A.2.1. The MLDRAM circuit Netlist 96

A.2.2. The Stimulus Template File 105

Appendix B: Test Vector Generation Program 110

Appendix C: Charge Injection Analysis 120

C.1 Charge Injection Theory 120

C.2 Multilevel Restore 121

C.3 Sign Sensing 125

C.4 Magnitude Sensing 131

Appendix D: Charge Injection Simulation 135

D.1 Charge Injection Analysis 135

D.2 Simulation 137

D.2.1. Overview 137

D.2.2. Setup 138

D.2.3. Measuring Important Capacitance Ratios 139

D.2.4. Multilevel Restore 142

D.2.5. Sign Sensing 147

D.2.6. Magnitude Sensing 148

D.3 Design Method to Eliminate Charge Injection Effects. 151

D.4 Other Sources of Signal Error 152

LIST OF TABLES

TABLE 2-1. Conversion function for Furuyama's MLDRAM. 27

TABLE 3-1. Test Chip Pads 47

TABLE 3-2. Power-up State 47

TABLE 4-1. Critical Timing Intervals 51

TABLE 5-1. Coding of Bits 60

TABLE 5-2. Reference generation for the left, center and right sub-bitlines 62

TABLE 5-3. Values of kj for each reference voltage when N = 7. 65

TABLE 6-1. Simulation Conditions 69

TABLE 6-2. VDC voltages used for simulating Furuyama's MLDRAM 72

TABLE 6-3. Timing Optimization Criteria 78

TABLE 6-4. Simulation Results 80

TABLE 6-5. MLDRAM Ranking 82

TABLE C-1. Charge sharing events affecting the voltage that is restored 122

TABLE C-2. Charge sharing events affecting sign sensing 125

TABLE C-3. Charge sharing events affecting magnitude sensing 131

TABLE D-1. Symbol definitions used in the analysis 136

TABLE D-2. Transistor sizes and gate voltages used for switching 138

TABLE D-3. Supply Voltages 138

TABLE D-4. Raw measurements from simulation 140

TABLE D-5. Ideal signal values 140

TABLE D-6. Measurements to determine K0 and K1 141

TABLE D-7. Raw measurements of charge injection taken from simulation 142

TABLE D-8. Final results from simulation of multilevel restore 143

TABLE D-9. Errors from capacitance imbalance due to CG of X0 145

TABLE D-10. Decomposition of the measured error at n = -2 146

TABLE D-11. Measurements taken from sign sensing simulation 147

TABLE D-12. Calculated errors in sign sensing 148

TABLE D-13. Final VSD components just before sign sensing. 148

TABLE D-14. Measurements taken from magnitude sensing 149

TABLE D-15. Calculated errors for magnitude sensing 150

TABLE D-16. Final VMD components just before sensing 151

LIST OF FIGURES

FIGURE 1.1. A conventional DRAM cell. 2

FIGURE 1.2. A DRAM core. 4

FIGURE 1.3. A two-bit-per-cell storage scheme. 6

FIGURE 2.1. A basic DRAM array. 11

FIGURE 2.2. (a) Open bitline, (b) folded bitline. 13

FIGURE 2.3. Twisted bitlines. 14

FIGURE 2.4. Capacitive coupling in twisted bitlines. 14

FIGURE 2.5. DRAM core circuitry. 16

FIGURE 2.6. A sense amplifier. 17

FIGURE 2.7. Hierarchy of a 256 Mbit DDR-SDRAM. 20

FIGURE 2.8. Schematic of a DRAM column. 21

FIGURE 2.9. Sense and restore timing for a DRAM core. 21

FIGURE 2.10. HDRAM macrocell block diagram. 23

FIGURE 2.11. (a) Conventional DRAM Cell [4], (b) HDRAM Cell. 23

FIGURE 2.12. Schematic for the Furuyama MLDRAM. 26

FIGURE 2.13. Timing for the Furuyama MLDRAM. 28

FIGURE 2.14. Schematic for the Gillingham MLDRAM. 29

FIGURE 2.15. Timing for the Gillingham MLDRAM. 31

FIGURE 2.16. Schematic for the Okuda MLDRAM. 33

FIGURE 2.17. Timing for Okuda MLDRAM. 34

FIGURE 3.1. MLDRAM chip floorplan. 37

FIGURE 3.2. Block diagram of the MLDRAM test chip. 37

FIGURE 3.3. Reference thresholds in sensing referred to the memory cell 39

FIGURE 3.4. Test point circuitry. 41

FIGURE 3.5. Example external circuitry for probing the bitlines. 41

FIGURE 3.6. Arrangement of the core components. 42

FIGURE 3.7. Wordline driver schematic 44

FIGURE 3.7. Example timing for turning a wordline on 44

FIGURE 3.8. De-scrambled row and column address map 45

FIGURE 3.9. Schematic of the Data I/O circuitry. 46

FIGURE 4.1. Control timing for the first MLDRAM chip 50

FIGURE 5.1. Sub-bitline designs. 57

FIGURE 5.2. Top level schematic of the new MLDRAM scheme. 58

FIGURE 5.3. Control timing for the new MLDRAM scheme 61

FIGURE 6.1. The model for four memory cells (4MC). 68

FIGURE 6.2. The model used for the sense amplifiers. 68

FIGURE 6.3. Schematic used for simulating Furuyama's MLDRAM [7]. 70

FIGURE 6.4. Schematic used for simulating Gillingham's MLDRAM [1]. 73

FIGURE 6.5. Schematic used for simulating Okuda's MLDRAM [8]. 74

FIGURE 6.6. Schematic used for simulating the proposed MLDRAM (Chapter 5). 77

FIGURE 6.7. Sub-bitline Schematics for the proposed MLDRAM (Chapter 5) 77

FIGURE 6.8. Sense and restore waveforms from the simulations 81

FIGURE D.1. Multilevel DRAM circuit configuration 136

GLOSSARY OF TECHNICAL TERMS

Charge Injection	When a transistor gate potential is changed, some charge is coupled from the gate to the source and drain via the parasitic gate-source capacitance and CGS gate-drain capacitance CGD.
Design Rules	The geometrical rules set by an integrated circuit manufacturer that define the minimum device sizes, device spacing, metal wire spacing as well as other critical dimensions.
Charge Pump	A circuit that uses a switched capacitor to generate a voltage outside the range of available voltage supplies provided from off chip. It is used on-chip to generate such voltages as the boosted wordline voltage.
Friendly Cell	A cell that is in the perimeter of the array and does not function as a memory cell. Friendly cells ensure that all functioning cells are neighbored by the same physical and electrical structure.
Grey code	A sequence of equal-length binary codeword that has the property that adjacent codewords differ by exactly one bit.
Cell Plate	The common electrical node of all storage capacitors in a DRAM device.

GLOSSARY OF ACRONYMS

CMC	Canadian Microelectronics Corporation.
CMOS	Complementary Metal Oxide Semiconductor, refers to a MOS process or circuit which has both n-type transistors (NMOS) and p-type transistors (PMOS).
DDR	Double Data Rate.
DRAM	Dynamic Random-Access Memory.
ESD	Electrostatic Discharge. Refers to the possibly damaging discharge of static electricity from the environment through the lead of an integrated circuit.
HDRAM	An embedded DRAM design produced by MOSAID Technologies Inc. that uses only logic process rules.
LSB	Least Significant Bit. This usually refers to the power of zero bit in a binary number.
MOS	Metal Oxide Semiconductor. Used as an abbreviation for MOSFET.
MOSFET	Metal Oxide Semiconductor Field Effect Transistor.
MSB	Most Significant Bit. This usual refers to the left-most bit position in a binary number.
NMOS	A MOS transistor that has n-doped source and drain regions sitting in a p-doped well.
PMOS	A MOS transistor that has p-doped source and drain regions sitting in an n-doped well.
SDRAM	Synchronous DRAM.
TSMC	Taiwan Semiconductor Manufacturing Corporation.
VBLP	Bitline Precharge Voltage. It is equal to one half VDD.
VCELL	The voltage of a DRAM memory cell storage node.
VCP	The cell plate voltage in a DRAM. This refers to the voltage source that connects to the cell plate node or the cell plate node itself.

VDD	This is the CMOS power supply voltage. Digital CMOS gates have the drain terminal of a PMOS device connected to the node with this label.
VSS	The source voltage for the NMOS devices in CMOS logic. This voltage is often used synonymously with ground or 0 V.

CHAPTER 1: INTRODUCTION

Memories are a vital part of most microelectronic digital systems, from small embedded controllers to large supercomputers. Memory performs the basic functions of retaining data and allowing this data to be retrieved by other components in a system. Semiconductor memory is a crucial part of digital systems and often accounts for a significant portion of the total system cost. The most widely used semiconductor memory type is DRAM. Annual sales of DRAM in 1998 were in the order of US $37 billion. Fierce competition among DRAM manufacturers drives the need to increase the storage capacity per chip and reduce the cost per bit. A technology that has not been successfully exploited commercially is multilevel DRAM (MLDRAM), which offers lower cost-per-bit without the expense of retooling the DRAM manufacturing process. MLDRAM differs from conventional DRAM by storing more than one bit per storage cell. This chapter introduces semiconductor memories including DRAM and MLDRAM. This thesis explores alternate MLDRAM circuit techniques and confronts the practical implementation issues of MLDRAM.

1.1 Semiconductor Memory

There are two basic types of semiconductor memories. Those that retain data after power is withdrawn are called nonvolatile memories and those that retain data only while power is applied are called volatile memories. Nonvolatile semiconductor memories that have permanent data are called Read-Only Memories (ROMs). Nonvolatile semiconductor memories that may be programmed after manufacture are known as Programmable Read-Only Memories (PROMs). Some nonvolatile memories can be erased and re-programmed, such as Erasable and Programmable Read-Only Memories (EPROMs), but such technologies can only be reprogrammed after the entire memory is erased. Flash memory is an electrically erasable and re-programmable ROM in which fractions of the full memory may be altered. Another nonvolatile memory to watch is ferroelectric RAM, which uses capacitors that exhibit hysteresis.

Volatile semiconductor memories are generally referred to as Random-Access Memories (RAMs). Specific memory locations within the memory can be read or written in a short time without affecting the other data in the memory. RAMs can be static (SRAM) or dynamic (DRAM). SRAMs use flip-flops to store the data while DRAMs store data as charge on capacitors. SRAMs can be designed to be very fast and so are used for high-speed embedded applications and cache memory. Unlike the high stability of SRAM cells, DRAM cells do not hold their voltages indefinitely as their storage capacitors leak charge to their surroundings. To prevent data loss, DRAMs require periodic "refresh operations" that regenerate the stored voltages. DRAMs are slower than SRAMs but they are the cheapest and most dense semiconductor memory technology. They have been the largest sales volume semiconductor memory type for the last two decades.

1.2 DRAM Technology

The modern DRAM cell consists of one transistor and one capacitor as shown in Figure 1.1. One bit of data is stored in the cell as one of two voltages on the capacitor (storage node). The voltages normally used for storage are V_{DD} (the chip supply voltage) and V_{SS} (ground).

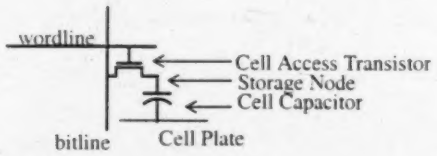


FIGURE 1.1. A conventional DRAM cell.

To write data into a cell, the bitline is driven to the corresponding voltage value and the wordline voltage is raised to turn on the cell access transistor (shown in Figure 1.1). This causes the storage node to be charged to the same voltage as the bitline. The wordline voltage is then lowered and the cell access transistor is turned off capturing the data voltage signal in the storage node. To read the data from the cell, the bitline is precharged to a midpoint voltage between the two allowed storage voltages. The midpoint voltage,

called the bitline precharge voltage (V_{BLP}), has the value $\frac{1}{2}V_{DD}$. With the bitline floating at the precharge voltage, the wordline voltage is raised and the cell access transistor is turned on to connect the storage node to the bitlines. The charge from the storage node causes a voltage change on the bitline. In typical DRAM processes, the cell capacitor ranges in capacitance from 25 fF to 60 fF while the bitline parasitic capacitance can be up to ten times the cell capacitance. Thus, when the cell capacitor is connected to the bitline, the voltage change on the bitline is only in the order of hundreds of millivolts. This small voltage change is detected by an amplifier, called a sense amplifier, that then determines what voltage, V_{DD} or V_{SS}, and hence what bit value, resided in the cell.

Figure 1.2 shows the organization of the DRAM cells in a two-dimensional array. Aligned along the rows are wordlines and along the columns are bitlines. The structure of the array uses a conventional "folded bitline" architecture (as explained in Section 2.2.1). Rather than have a cell at every intersection of wordline and bitline, the cells are provided at only half of the intersections so that any one wordline controls cells connected to every other bitline. This organization greatly increases noise rejection in the sense amplifier.

The key internal operations in DRAMs involve sensing and restoring cell voltages. The sensing and restoring techniques for conventional two-level DRAM are well known and do not vary much between designs [4]. The sense amplifiers behave as differential amplifiers that compare the voltages of two adjacent bitlines. Since the cell access transistors are only present at alternate bitlines, one bitline will remain precharged to V_{BLP} while the adjacent bitline will have a voltage change determined by the data from the activated cell. By comparing the voltage of the two bitlines the data is read from the cell.

The basic structure of a DRAM sense amplifier is two CMOS inverters connected in a ring. These special inverters may be connected to and disconnected from the power supplies. For sensing, the sense amplifier starts off unpowered and with its outputs shorted together and precharged to V_{BLP}. Then, the shorting path between the outputs is disconnected and the inputs to the inverters are connected to the bitline pair. The inverters

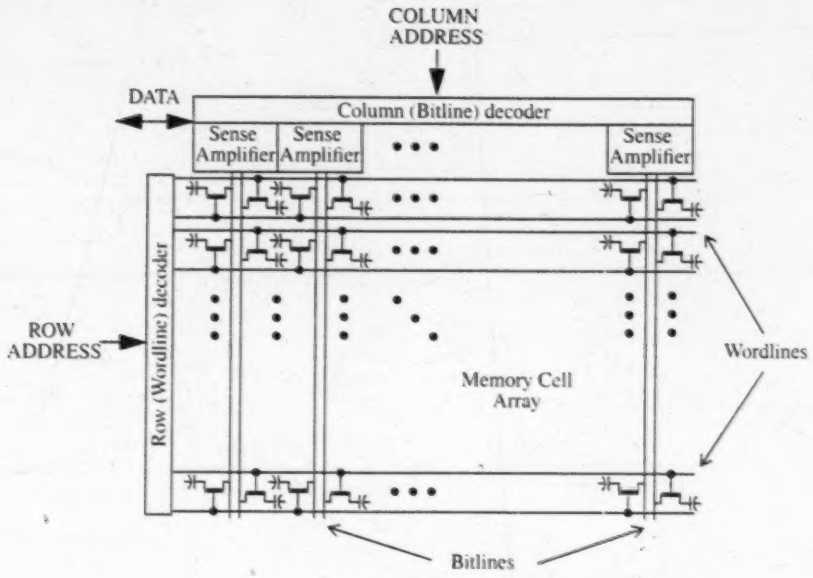


FIGURE 1.2. A DRAM core.

are then powered up and act together as a differential amplifier with positive feedback. The bitlines are quickly charged to opposing power supply voltages as determined by the differential input signal. While activated, the sense amplifier acts as a CMOS latch that holds the bit value from the accessed cell. This latched value may be read off-chip, or overwritten with new data, before it is restored back to the memory cell.

1.3 MLDRAM Technology

It is becoming increasingly expensive to raise the storage density of DRAM by reducing the physical size of the cells and using even more complex three-dimensional cell capacitor structures. Density innovations in DRAM processes are approaching economic limits in terms of the cost of manufacturing [5]. The demand for increasing bit density may soon be insufficient to make processing innovations cost-effective. One additional dimension, which has yet to be successfully exploited in commercial parts, is to store more than one bit per cell. In a DRAM cell this involves storing and then

subsequently sensing more than two distinct voltage levels on the cell capacitor. In DRAM, the cell voltage can be defined as one of two possible values (Eq. 1.1).

$$V_{CELL} \in \{a_0, a_1\}(V_{DD}), \text{ where } 0 \le a_i \le 1 \tag{1.1}$$

In MLDRAM, the number of allowed states, N, is greater than two. N is related to the number of bits (n) by Eq. 1.2.

$$N = 2^n \tag{1.2}$$

Thus, for n-bit storage the allowed cell voltages levels are:

$$V_{CELL} \in \{a_0, a_1, \ldots, a_{N-1}\}(V_{DD}) \tag{1.3}$$

The coefficients a_i should be chosen to maximize the noise margins between them. This can be achieved if the allowed voltages in the cell are equally spaced in the available voltage range as follows:

$$V_{CELL} \in \{0, 1, 2, \ldots, N-1\}\frac{V_{DD}}{N-1} \tag{1.4}$$

To read the multilevel data, the cell voltage is compared to a set of reference voltages. There is one reference voltage midway between each allowed cell voltage. The N-1 reference voltages (V_{REF}) are equally spaced between the possible cell voltages (V_{CELL}) to give equal and maximum noise margins between all references and cell voltages. The reference voltages are given in Eq. 1.5.

$$V_{REF} \in \{1, 3, 5, \ldots, 2N-3\}\frac{V_{DD}}{2(N-1)} \tag{1.5}$$

Although any number of bits may be stored in each cell, the number chosen in most MLDRAM designs is n=2 [1,7,8].

Figure 1.3 illustrates how one might encode the four logic pairs 00, 01, 10, and 11 as four equally-spaced voltage levels in the range V_{SS}=0 to V_{DD}. The cell voltage must be compared to the reference levels to extract the two bits from a cell. If the cell voltage is less than $\frac{1}{6}V_{DD}$ (the midpoint voltage between 0 and $\frac{1}{3}V_{DD}$) then the value of the bits is 00. If the cell voltage is greater than $\frac{1}{6}V_{DD}$ and less than $\frac{1}{2}V_{DD}$ then the value of the bits is 01, and so on.

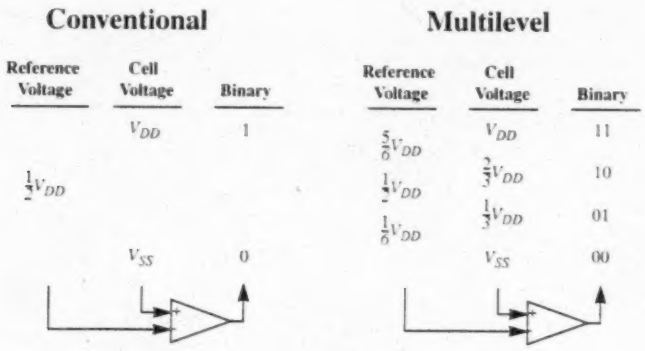


FIGURE 1.3. A two-bit-per-cell storage scheme.

One important challenge in MLDRAM that is evident in Figure 1.3 is that the differential signal strength, and thus the noise margins for a 4-level MLDRAM, will be only one third those of a conventional 2-level DRAM with the same supply voltage. In general, an N-level MLDRAM has noise margins reduced by $\frac{1}{N-1}$ over 2-level DRAM. This makes MLDRAM more vulnerable to soft errors due to leakage current from the cells, sense amplifier offsets and environmental radiation such as α-particles. It is still unclear whether or not the advantages of MLDRAM can be achieved despite this technical challenge.

When implementing MLDRAM, it is the sense and restore scheme that becomes the most difficult circuit to design. The multilevel sense and restore scheme must provide the capability to extract data encoded as one of the many allowed voltage ranges on a cell

capacitor (the sense operation), and have the capability to take multiple bits and convert them to one of many nominal voltage levels to be placed on the cell capacitor (the restore operation). In essence, sensing is analog-to-digital conversion and restoring is digital-to-analog conversion. The fundamental problem in MLDRAM design is to find a circuit that performs these operations within the small area available in the pitch of one, two or four bitlines.

Once a suitable circuit is found, it must also satisfy a number of other criteria for it to function reliably. One criterion that must be satisfied is that the overall area of the sense and restore circuitry should be minimal. MLDRAM can double the storage density of the cell array but this gain is reduced by any addition of area overhead in the sense amplifiers and chip periphery.

With a satisfactory MLDRAM sense and restore scheme (i.e. one that fits in the bitline pitch and does not add too much area overhead) the following criteria become important to determine the feasibility of the design. First, the design should be insensitive to IC processing deviations. Processing variations can cause circuit parameters to change disproportionately. Circuits should be designed with some degree of robustness against inevitable small processing variations. Second, noise insensitivity is required. DRAMs are electrically balanced so that the common-mode noise rejection in the sense amplifiers is inherently high. This requirement becomes even more important in MLDRAM because, as noted earlier, noise margins are decreased considerably. Thus, great care is required to compensate for or avoid signal degradation due to such factors as charge injection, sense amplifier offsets and noise coupling from elsewhere in the circuit. Third, it is desirable that MLDRAM be comparable in speed to DRAM. This requirement is difficult to meet because of the potential increase in sequential steps needed to extract and then write back multilevel encoded data. With innovations in DRAM organization and interfaces, it is conceivable that a small time increase would be tolerable and slightly slower MLDRAM random access times would still be competitive with DRAM. Should a circuit require too long to perform sense and restore, it may not compete as a substitute for DRAM and so

may only succeed in other smaller niche markets such as file store memories, where fast random access is not a requirement.

1.4 Thesis Outline

Many MLDRAM circuit designs have been proposed in the literature. Most have been implemented on large test chips but none have entered commercial production. As well, there are unimplemented designs that have not been fully studied. I believe that it is possible to design an MLDRAM robust enough for economical manufacture. This thesis explores the feasibility of MLDRAM as a step towards finding a reliable and economically manufacturable design.

The next chapter introduces the concepts of semiconductor memory and DRAM. Useful DRAM design techniques, terminology and basic circuits are explained in order to facilitate discussions of the design challenges and solutions of MLDRAM implementation. There is an overview of the basic concepts of MLDRAM and a review of past MLDRAM designs. MLDRAM benefits, drawbacks, and technical challenges are summarized.

Most of the MLDRAM schemes in the literature have been implemented and tested. A recent MLDRAM design proposed by Gillingham [1] has not been implemented and characterized. Chapters 3 and 4 are devoted to exploring the implementation of this MLDRAM scheme with the goal of demonstrating that it works in practice. In Chapter 3 the design details of an MLDRAM test chip are presented. The test equipment, test methodology, and test results from the MLDRAM test chip are described in Chapter 4.

It is evident from the literature that there are many ways to implement MLDRAM, each of which has different strengths and weaknesses. One of the strengths that seems to have been poorly addressed in past designs, with the exception of Gillingham's scheme, is robustness. Robustness as a characteristic is difficult to quantify as it takes many forms. In general, though, it is the robustness of the design that will determine if usable parts can be manufactured. Gillingham addresses this issue in great detail yet his scheme compromises

speed. Chapter 5 discussed a new MLDRAM scheme that attempts to draw on the strength of Gillingham's robust design while overcoming issues of speed. A description of the circuit design is given as well as the circuit operation. There is a brief discussion of the strengths and weaknesses of the new MLDRAM scheme at the end of the chapter.

With all of the past designs described in theory in separate papers in terms of different DRAM processes and test chip implementations (if any), it is difficult to compare the schemes with one-another quantitatively. Chapter 6 presents a simulation study that attempts to compare three past designs and the design proposed in Chapter 5. The approach taken was to create a hypothetical DRAM process that resembles the current technology and to simulate each of the schemes as they would be implemented in this hypothetical process. Careful attention is paid to ensuring a fair comparison. Parameters that are independent of the MLDRAM design, such as circuit parasitics, circuit sizes, and basic DRAM core elements, are kept constant across all simulation models. The circuits are characterized and compared for access time, cycle time, and cycle energy-per-bit. As well, a discussion of the qualitative aspects of area overhead, robustness, and complexity is presented.

A detailed analysis of charge injection effects in Gillingham's MLDRAM is provided in Appendix C.

CHAPTER 2: DRAM AND MLDRAM

This chapter provides background information about DRAM architectures and circuit design techniques. The techniques presented were used in the design of an MLDRAM test chip (described in Chapter 3). The architecture of the test chip is based on the pre-existing HDRAM macrocell from MOSAID Technologies Inc., which is described in Section 2.5. A review of past work in MLDRAM is given in Section 2.6. The chapter concludes with a brief discussion of the advantages and disadvantages of each prior MLDRAM.

2.1 DRAM Manufacturing Processes

To understand DRAM circuitry it is important to understand the unique aspects of DRAM process design rules. There are three main parts of a DRAM circuit distinguishable by function and by design rules. They are the array, the core (not including the array) and the periphery. The array refers to the memory cell array. In this region, special processing steps are used for the formation of cell capacitors, cell access transistors and bitline and wordline pitch. The design rules in the array often do not include PMOS devices and may include multiple polysilicon layers to form the bitlines, storage nodes and cell plate node (see Figure 1.1 on page 2). An array augmented with immediately surrounding circuitry is called a core. The additional circuitry in the core contains the sense amplifiers, precharge devices, wordline drivers, and at least some of the address decoding logic. Core circuitry is constrained to fit in the pitch of only a few bitlines or wordlines. Key processing steps for the core are optimized for these circuits so design rules in the core may allow smaller and more tightly spaced devices than those in the periphery. The periphery is the area of the chip between and around the cores (there are typically many cores) that contains all of the interface circuitry, power regulation circuits, pads, and timing control logic.

2.2 Basic DRAM Circuits and Design Techniques

2.2.1. The DRAM Array

A DRAM array is a regular two-dimensional array of DRAM memory cells, as shown in Figure 2.1. All of the wordlines run in one direction and all of the bitlines run in the other. The array has a folded bitline architecture (explained below) so there is a memory cell at every second intersection between bitlines and wordlines. Many design techniques may be incorporated into the basic array of Figure 2.1. The most common techniques are explained below.

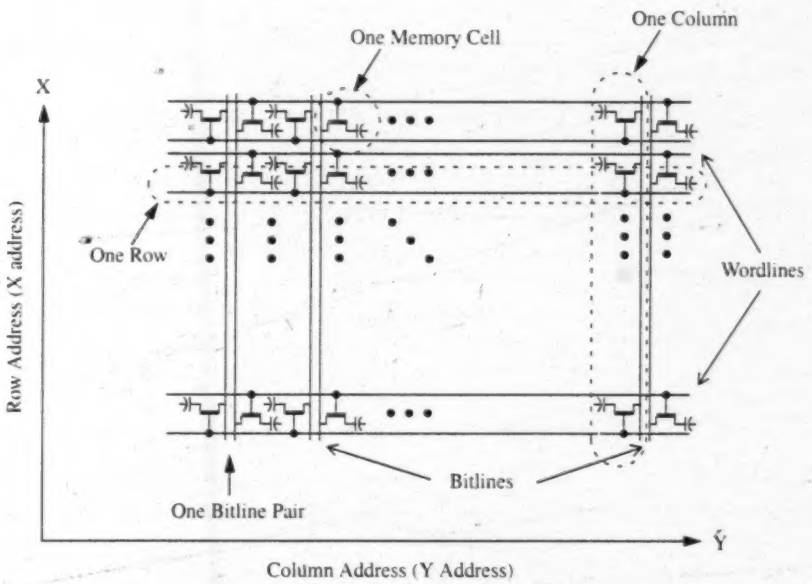


FIGURE 2.1. A basic DRAM array.

Rows and Columns

Positions in the array are given in terms of rows and columns. The term "row" refers to the address of a particular wordline. The symbol "X" is used to denote the address of a row. The terms "row" and "wordline" are sometimes used interchangeably when describing the array although "wordline" refers to the electrical node that runs along the length of that row. When a particular row is accessed, the wordline in that row and all of

the cell access transistors connected to it are activated. The term "column" refers to one or more bitlines that are all accessed using the same column address. The symbol "Y" is used to denote the address of a column. Usually "columns" and "bitlines" can be used interchangeably, but this can be confusing sometimes because a column may contain two or more bitlines (or bitline pairs, explained below).

Friendly Cells

One assumption in DRAM design is that all of the cells have the same size, capacitance, electrical parameters, and surrounding electrical environment. In reality there is some variation in these parameters, but it is desirable to minimize the variation whenever possible. At the edge of the array there is a discontinuity in the topographical composition of the processed die that can affect the quality of the perimeter cells. As well as the physical discontinuity, there is parasitic capacitance discontinuity because all cells, except those at the edge of the array, are surrounded by identical memory cells. It is therefore common practice to add "friendly cells" all around the perimeter of the array. Friendly cells are never used for data storage; in fact the gates of all friendly cell access transistors are permanently connected to a power supply (V_{SS} for NMOS).

Folded Bitlines and Twisted Bitlines

Because DRAM cell capacitance is much less than the bitline parasitic capacitance, the attenuated voltage signal that the sense amplifier must detect is in the order of a few hundred millivolts. This voltage may be further reduced if the cell data signal was already degraded by leakage currents. The small voltages involved make the sensing operation vulnerable to errors caused by noise and component mismatch. Two common ways to combat noise on the bitlines are to use both folded bitlines and twisted bitlines.

Folded bitlines, shown in Figure 2.2, are also referred to as bitline pairs. Any external noise coupled into bitline BL will be almost equally coupled into $\overline{BL}$ because the two bitlines are parallel and very close together. Such noise is readily rejected by a differential mode sense amplifier. The close proximity also keeps the electrical parameters of the pair nearly equal despite any process gradients that may exist across the array. The open bitline

configuration, which has a cell at every intersection of wordline and bitline, was used until the 64 Kb generation. The open bitline has a higher storage cell density than the folded bitline, but it is susceptible to noise and parameter errors because the bitline BL0 is not in the proximity of the reference bitline BL1.

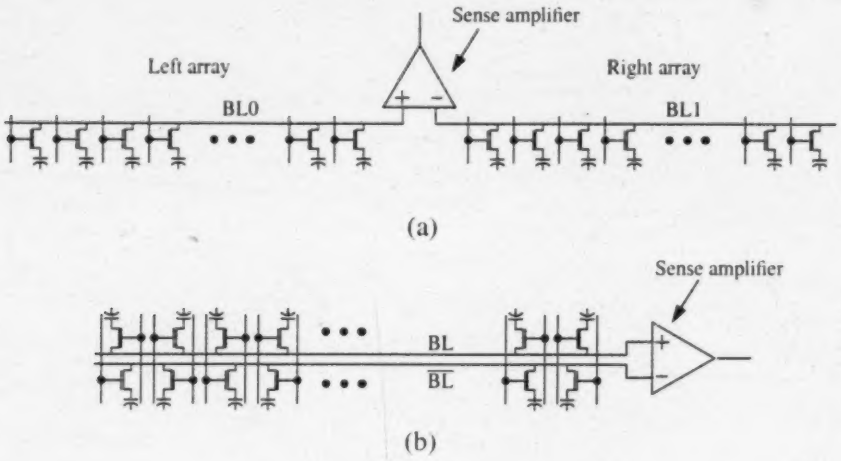


FIGURE 2.2. (a) Open bitline, (b) folded bitline.

Another technique for noise suppression is the use of twisted bitlines. With the folded bitline configuration, the bitlines can be twisted once or twice so that the folded bitline becomes a twisted pair transmission line. An example of a twisting scheme is shown in Figure 2.3. The twisting scheme has three twist regions. Every second bitline is twisted once while the others are twisted twice. Figure 2.4 shows the parasitic capacitances of adjacent bitlines for folded and twisted bitlines. In the folded bitline, there is unequal coupling between bitlines in a pair because $\overline{BL0}$ and BL1 are side-by-side along their entire length. Thus a disturbance on $\overline{BL0}$ will cause a larger disturbance on BL1 than on $\overline{BL1}$. For the twisted bitlines, however, a disturbance on $\overline{BL0}$ will cause an equal disturbance on each of BL1 and $\overline{BL1}$. Such noise is again readily rejected by the sense amplifier across BL1 and $\overline{BL1}$.

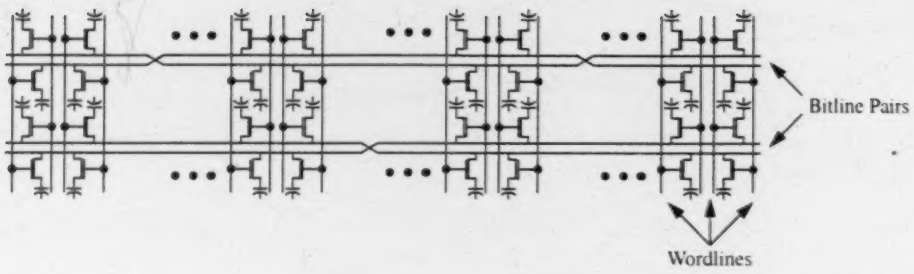


FIGURE 2.3. Twisted bitlines.

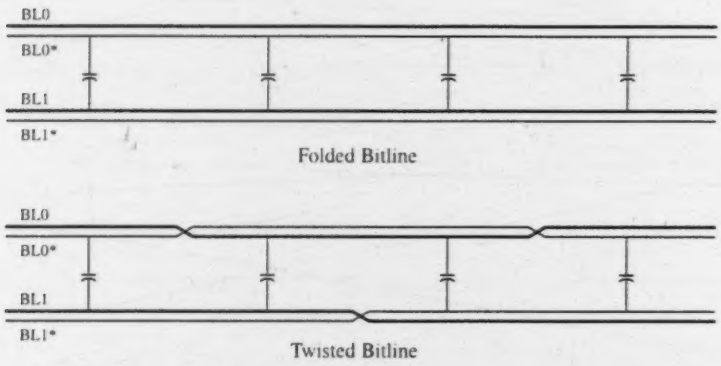


FIGURE 2.4. Capacitive coupling in twisted bitlines.

Dummy Cells

In DRAM sensing, the sense amplifier compares the attenuated signal from the memory cell with the precharge voltage V_{BLP}. This is because both bitlines are precharged to the same voltage V_{BLP} prior to a cell access. When a cell is accessed, only one of the bitlines is charge shared with a memory cell. The other bitline in the pair remains at V_{BLP}. Thus the sensing outcome depends upon the means used to produce a reliable V_{BLP} voltage.

Instead of relying on an accurate bitline precharge value, the data in the cell can be compared to a reference voltage stored in a special cell, called a dummy cell (or a

reference cell). Referring to Figure 2.2(b) above, when a memory cell connected to BL is accessed, the dummy cell connected to $\overline{BL}$ is accessed. The bitlines, having started with the same voltage, now experience a differential voltage equal to the difference in charge between the memory cell and the dummy cell. The sensing operation does not depend on the precharge voltage, rather, it depends on the voltage stored on the dummy cell capacitor, $\frac{1}{2}V_{DD}$.

Use of dummy cells is especially valuable for DRAMs that use a precharge voltage other than $V_{BLP} = \frac{1}{2}V_{DD}$. But, regardless of the particular precharge voltage, there is another advantage to using dummy cells. When accessing a cell the small inevitable charge injection across the gate-to-drain capacitance of the cell access transistor causes a small error in the final voltage on the bitline. By using a dummy cell, the same charge is injected by the dummy wordline that controls the dummy cells connected to each of BL and $\overline{BL}$. Thus the error caused by charge injection can be cancelled out to a first order approximation. The drawback to using dummy cells is that an accurate reference voltage must be stored in the dummy cell prior to each cell access, which may require extra circuitry and may add time to the access cycle.

2.2.2. The DRAM Core

The DRAM core contains all the circuitry that is within the bitline or wordline pitch. In the wordline pitch there are wordline drivers and row address decoders. In the bitline pitch there are sense amplifiers and column address decoders. The core circuitry components are shown in Figure 2.5. Not shown in Figure 2.5 are additional rows and/or columns of spare cells that are typically used to increase the effective yield via on-chip repair, a technique that is called static redundancy.

Row Decoders and Column Decoders

Both the row and column decoders have the same basic structure. For each, address decoding is split into two stages. The first stage, predecoding, is done in the periphery. The predecoder takes groups of two address bits and supplies each of the four minterms to the

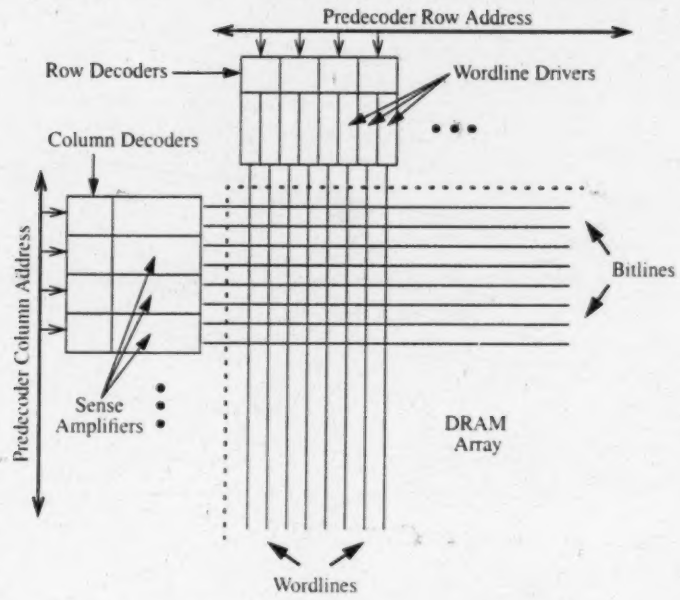


FIGURE 2.5. DRAM core circuitry.

decoder. For example, the first two address bits A_1 and A_0 are predecoded into the four minterms $\overline{A}_1\overline{A}_0$, $\overline{A}_1A_0$, $A_1\overline{A}_0$, and A_1A_0. The final decoding for the row can then be done in the pitch of a wordline by logically operating on only one of the four minterms from each predecoded pair. These second-stage decoders are specially designed to fit in the pitch of one or more wordlines. Similarly, final column decoding is done in the column pitch, which is the pitch of one or more bitline pairs.

Wordline drivers

Wordline drivers are required to drive the relatively large capacitive load of the wordlines, which may include the contributions of thousands of cell access transistor gates. In addition, the wordline drivers may be required to drive the wordline above V_{DD} (called wordline boosting). This is needed to allow the full V_{DD} voltage to be passed into the cell storage node. (Signals of amplitude V_{DD} are attenuated when they are conducted

through the channel of an NMOS transistor.) However, wordline boosting is not always employed due to the reduced long-term reliability of the cell access transistors.

Sense Amplifiers

A typical sense amplifier circuit is shown in Figure 2.6. The four transistors M_{P1}, M_{N1}, M_{P2}, and M_{N2} form two cross-coupled CMOS inverters and are responsible for bitline signal amplification. The M_S and M_R transistors are used to enable and disable sensing by controlling the connections to the power supply rails. Transistors M_{PRE1}, M_{PRE2}, M_{PRE3}, M_{PRE4}, and M_{PRE5} connect the sense amplifier and the bitlines to the precharge voltage V_{BLP}. While the bitlines are precharged, signals R and S keep the sense amplifier disconnected from the power supplies. Just before a wordline is activated, the precharge devices are deactivated by de-asserting signal EQL. This leaves the bitlines and the sense amplifier nodes floating. After the charge from the accessed cell is shared with one of the bitlines, the sense amplifier is turned on by asserting R and S, and the data is sensed and amplified up to a signal that reaches the power supply rails. Now, the sense amplifier acts as a CMOS latch. One of the bitlines is charged to V_{DD} and the other to V_{SS}. At this point the data may be read out onto the data bus through M_{Y1} and M_{Y2} or the sense amplifier may be overwritten from the data bus. This assumes that the corresponding COLUMN-SELECT signal has been asserted by the column decoder. With the bitlines fully charged, the wordline is deactivated and data signal is captured inside the same cell that was previously read.

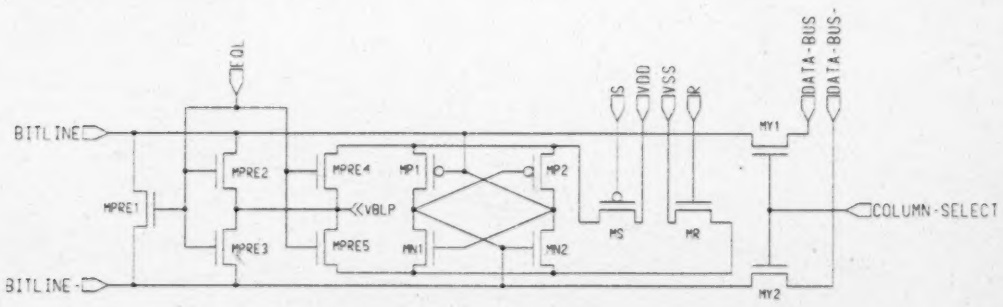


FIGURE 2.6. A sense amplifier.

Data Bus

The data bus usually runs over or beside the sense amplifier and may be many bits wide. The bitline sense amplifiers are quite small and are not able to drive the long, highly capacitive data bus to V_{DD} and V_{SS} in a reasonable time. To speed up data transfers over the data bus, the data bus uses differential signals and is kept at V_{BLP} when idle. When a sense amplifier is selected by asserting the COLUMN-SELECT signal (referring to Figure 2.6), the weak drive of the sense amplifier can quickly cause a small differential signal on the data bus pair of wires. This differential signal is detected in the periphery by large data bus sense amplifiers. As with the bitline sense amplifiers, the data bus sense amplifiers are vulnerable to errors from noise. Twisting the data bus wires in a fashion similar to twisted bitlines helps the data bus sense amplifier reject noise that may be induced onto the data bus by surrounding signal transitions.

Static Redundancy

Commercial DRAMs must provide fully functional RAM capabilities for each memory address. Often in manufacturing there is a small defect on the chip that prevents a cell, a small group of cells, a row (wordline), or a column (bitline) from functioning correctly. Without the ability to work around such localized defects, the entire IC would have to be discarded. Static redundancy is built into the core of the chip by adding extra rows of cells (redundant rows) and extra columns of cells (redundant columns). These redundant rows and columns are not normally used except in the case that a small defect prevents the use of one row or column in the array. In that case, the row or column decoder is modified permanently through fuses so that the bad row or column is never accessed and one of the redundant rows or columns is accessed instead for the same address. This technique directly improves manufacturing yield because it allows faulty chips to be "fixed" and sold rather than discarded. However, there is an area penalty paid when adding redundancy. The number of redundant rows and columns added, and thus the number of faulty IC's that can be fixed, must be balanced with the overall cost increase of each IC due to added area. Typically at least two redundant rows and two redundant columns are provided in each core to allow shorts between bitlines or wordlines to be repaired.

2.2.3. The DRAM Periphery

The DRAM periphery contains all of the rest of the chip circuitry. This includes the row and column predecoders, address latches, charge pumps for wordline boosting, voltage dividers for V_{BLP} generation and regulation, data bus sense amplifiers and write drivers, and timing control for the core and interface.

2.3 DRAM Integrated Circuits

DRAM Hierarchy

There are practical limitations to the size that a DRAM core can be. The longer the wordline the longer the access time. Longer bitlines mean larger bitline capacitances and weaker signals. In large ICs compromises are made between core size and the number of cores used. Mega- and Giga-bit DRAMs typically have many arrays arranged into banks of arrays. The core circuitry is split between the arrays and the bank. The bank as a whole has one row decoder and one column decoder. Each array has its own sub-wordline drivers. Main wordline drivers along the edge of the bank drive wordline signals to each sub-wordline driver in the bank. The sub-wordline driver then drives the cells in the array. An example of a typical 256 Mbit DRAM that uses these techniques is shown in Figure 2.7. The 256 M-bits are divided into 8 banks of 32 Mbits and each bank has 256 arrays of 128 Kbits each.

DRAM IC Parameters: Cell Efficiency and Die Efficiency

Two parameters that are commonly used to gauge a DRAM design are cell efficiency and die efficiency. Cell efficiency is defined as the die area required per bit in the cell array. For regular DRAM, this is equal to the area of the memory cell. Die efficiency is the area required per bit on average for the IC. It is calculated by dividing the total area of the memory chip (including all forms of overhead) by its total memory capacity.

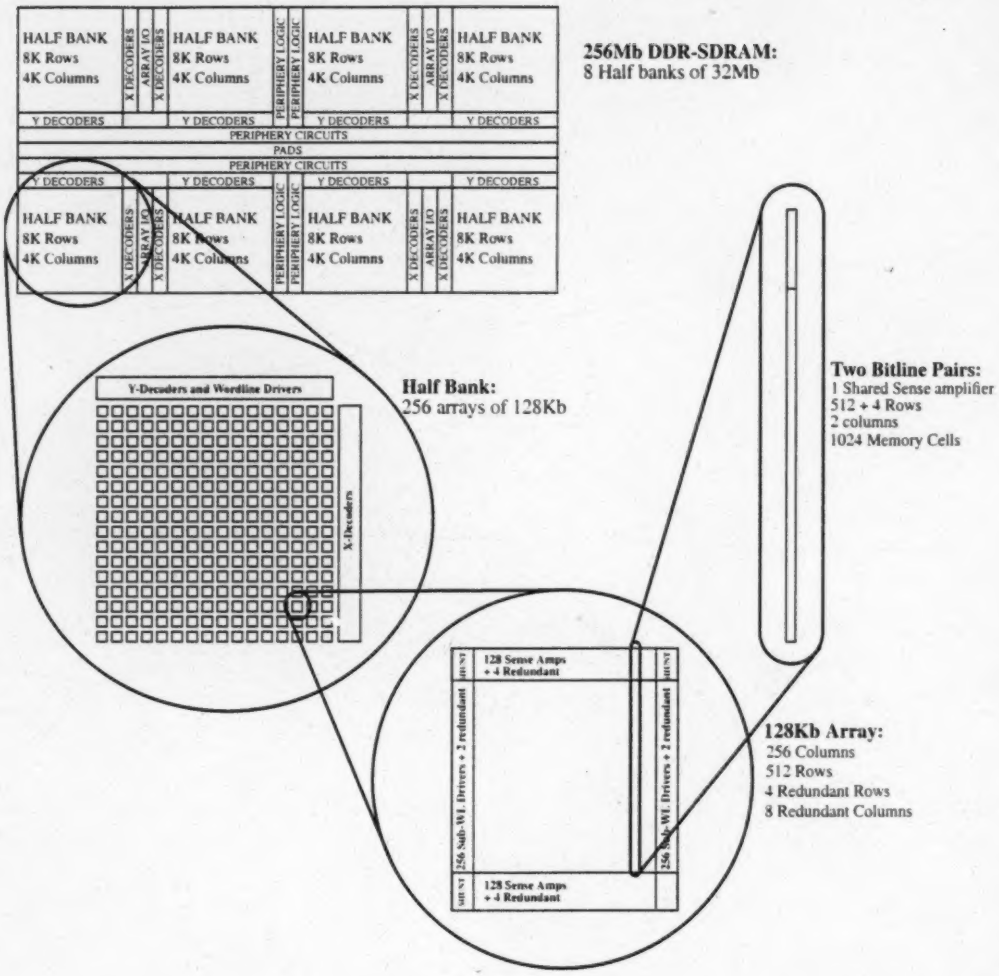


FIGURE 2.7. Hierarchy of a 256 Mbit DDR-SDRAM.

2.4 DRAM Operation

A basic DRAM bitline pair and sense amplifier are shown in Figure 2.8. The sense amplifier may be connected to the bitline pair by activating the signal I. The folded bitline pair is shown with two wordlines, Wi and Wi+1, to represent any pair of adjacent

wordlines in the array. Also shown are column access devices (activated by signal Y) that connect the sense amplifier to the differential data bus DB and $\overline{DB}$.

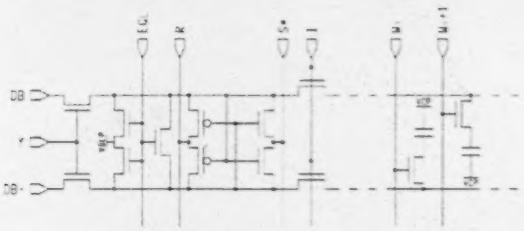


FIGURE 2.8. Schematic of a DRAM column.

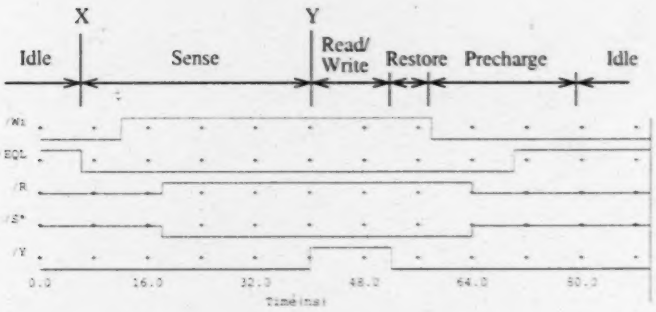


FIGURE 2.9. Sense and restore timing for a DRAM core.

A basic DRAM sense and restore cycle, shown in Figure 2.9, begins in the idle state where all of the bitlines and the sense amplifiers are precharged to V_{BLP}. The switching off of the I signal is not shown as it is not needed for the most basic DRAM core configuration. Signal I is used in some DRAMs to allow sense amplifier sharing between adjacent arrays. When a row address is supplied to the chip (indicated by the X in Figure 2.9) the precharge devices are switched off and the wordline is switched on. Next, the sense amplifier is activated through R and $\overline{S}$ (which are power supplies in this case, unlike the R and $\overline{S}$ controls shown in Figure 2.6) causing the bitline to which the wordline is connected to be compared to the other bitline, which still retains the precharge voltage V_{BLP}. After a short time, the sensing operation is complete and a column read or write operation may be performed. Application of a column address (indicated by the Y in

Figure 2.9) causes one or more of the Y signals to be driven high. With the sense amplifier connected to the data bus, the bit value on the sense amplifier may be read via the main sense amplifier. Alternatively, the main data bus amplifier can be used to force the state of the local sense amplifier as part of a write operation. After all column operations are complete, the data is restored by lowering the wordline to capture the data in the cell. Finally, the bitlines are returned to the precharge state and remain idle until the next cycle.

2.5 Embedded DRAMs

Embedded RAMs are memories that are integrated with other complex digital systems. They include so-called application-specific, memory-on-logic, and system-on-a-chip designs. Embedded SRAMs have been used in logic chips because the basic SRAM latch is naturally compatible with digital logic processes. Embedded DRAMs are becoming more popular, despite the compromises that must be made to reconcile DRAM and logic processes, as they offer the higher density required for high performance architectures [12].

2.5.1. The HDRAM Macrocell

A proven MOSAID design called HDRAM is an embedded memory for ASIC applications [10-11] that was used for the design of an MLDRAM test chip (Chapter 3). The next section is a brief overview of the HDRAM architecture.

HDRAM is available in a digital logic process so there are no special processing steps or design rules. The macrocell, shown in Figure 2.10, consists of two 128 Kbit blocks. The two blocks share column decoding circuitry but are otherwise independent memories. The macrocell contains no pad structures since it is an embedded memory. o

The HDRAM memory cells are PMOS transistors rather than capacitors (Figure 2.11). The cell storage capacitance here is the MOSFET parasitic gate-to-drain/source capacitance. PMOS is used rather than NMOS because PMOS devices can be electrically isolated from the rest of the chip. For maximal and linear storage capacitance, the transistor must operate in the so-called triode region [3]. For triode operation the cell plate

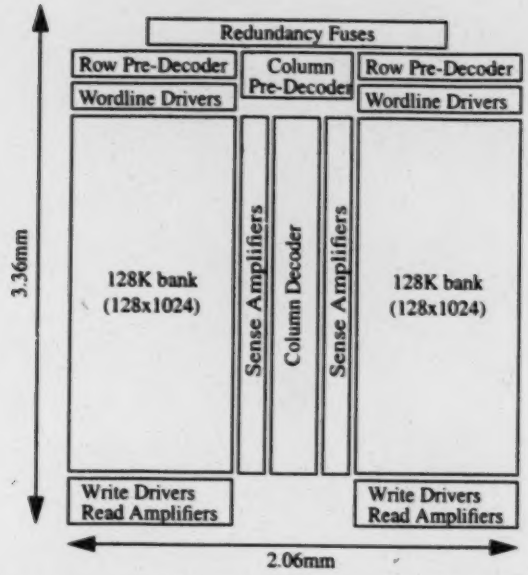


FIGURE 2.10. HDRAM macrocell block diagram.

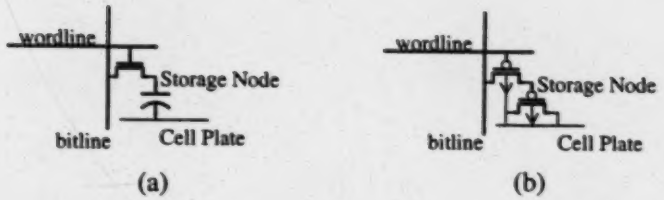


FIGURE 2.11. (a) Conventional DRAM Cell [4], (b) HDRAM Cell.

voltage must be more than one threshold voltage ($|V_{tp}|$) above the largest allowed storage voltage, V_{DD}=3.3V.

2.6 MLDRAM Methods and Circuits

MLDRAM circuit techniques generally involve modifying a DRAM core by adding circuitry and using new control sequences. M. Aoki *et al.* designed a 4-bit-per-cell DRAM that involves placing a rising staircase on the wordline and detecting when the bitline differential is disturbed [6]. Other schemes followed but they have all used a 2-bit-per-cell approach [7,1,8]. T. Furuyama *et al.* proposed a scheme that has the bitlines divided into

three equal-length sections, each with its own sense amplifier [7]. Others have taken a similar approach, each proposing different changes to the basic folded bitline structure [1][8]. The more recent schemes keep the wordline driver circuitry, array layout and memory cell operation the same as in DRAM. The following sections outline the changes made to a DRAM core in past MLDRAM designs.

The focus on bitline components for MLDRAM stems from the need to sense and restore multiple voltages quickly. The core column components (sense amplifiers, isolation devices, folded bitlines, dummy cells) are all well understood and proven circuits. A key limitation to MLDRAM circuitry is that it must fit in the bitline pitch, so it is not surprising that the proposed schemes are mainly composed of the circuit structures that are already optimized to fit in the column.

With modifications to the core come modifications to the basic control timing internal to an MLDRAM chip. The proposed MLDRAM schemes generally have more control signals and more total signal edges (i.e. signal transitions) than DRAM. Modifications to the internal chip control can achieve the required core timing and thus any MLDRAM scheme could be operated using the appropriate control timing to function as if it were a DRAM. This is possible because modifications to the core may be hidden from the user of the IC as the basic functionality of the chip as a dynamic semiconductor memory remains the same.

2.6.1. Aoki's MLDRAM

M. Aoki *et al.* propose a 4-bit per cell multilevel sense and restore technique in [6]. New sense and restore circuitry is added in place of the regular bitline sense amplifier. The basic operation of the scheme is timed to a staircase pulse that is applied to the selected wordline.

Data to be written is stored in a special register in the column. Data is written into the cell as one of sixteen voltage levels by synchronizing the staircase wordline signal to the write circuitry controlled by the column register. When the wordlines begin descending, the bitlines are set to 0 Volts. The column register, carrying a data value i, causes the

bitline to rise from 0 V to V_{DD} during the *i*-th step of the descending wordline voltage. The n-channel cell access transistor allows current to flow from the raised bitline into the cell until the cell voltage is equal to one threshold voltage below the wordline voltage. Thus one of sixteen levels is chosen by raising the bitline voltage during the appropriate step interval of the descending wordline.

Data is read from the cell by applying an ascending staircase to the wordline. For a cell that contains voltage level *i*, the bitline will first be disturbed when the wordline reaches voltage level *i*+1. To sense the small voltage, pre-amplifiers are placed in the bitline pitch. The outputs of the pre-amplifiers are fed to a bitline sense amplifier. The time when the sense amplifier detects the first disturbance on the bitline determines the read value according to the current state of the wordline staircase clock. Data may be read or written to the column register prior to the write operation, which restores the correct signal voltage to the cell.

This scheme has the advantage of being scalable to any number of levels. The number of levels is determined by choosing the number of steps that the wordline will take in its ascent and descent and the rest of the supporting circuitry can be scaled accordingly. There is a limit, however, to the number of levels that can be stored. When the charge stored per level becomes comparable to the charge lost through sub-threshold leakage from the cells, erroneous sensing results. The main disadvantage of this scheme is that it is slow. The read operation requires sixteen sequential sensing operations and the write operation requires the application of a slightly faster, but still relatively slow, 16 level staircase. As a result, this MLDRAM is probably restricted to use as a sequential-access mode file memory. However, this memory design is a precursor to recent MLDRAMs and thus serves as an important starting point.

2.6.2. Furuyama's MLDRAM

T. Furuyama *et al.* propose a multilevel sense and restore method in [7]. The sense and restore circuitry is shown in Figure 2.12. The bitline is split into three equal sections, called sub-bitlines, that are separated by switches controlled by SWT. Each sub-bitline can

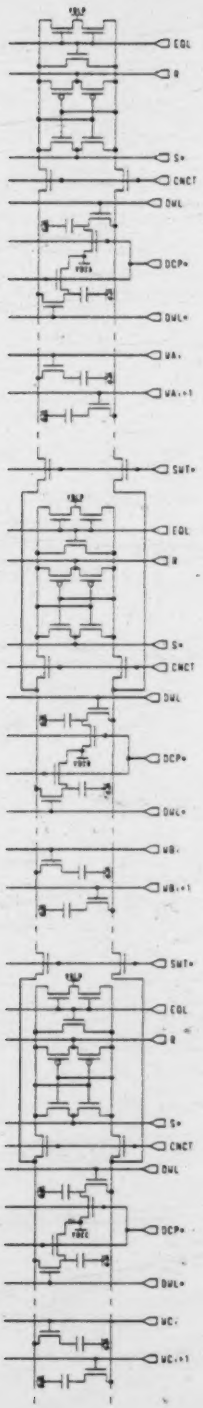


FIGURE 2.12. Schematic for the Furuyama MLDRAM.

be connected to a regular sense amplifier through switches controlled by CNCT. In addition to these bitline modifications, the cell array is modified so that each sub-bitline has one dummy cell. These dummy cells have two transistors rather than one. One transistor connects the dummy cell node to the bitline while the other connects the dummy cell node to one of the voltage supplies VDCA, VDCB, or VDCC. From left to right, the connections in the dummy cells are to VDCA for the first sub-bitline, VDCB for the second, and VDCC for the third. The three supplies are set to the three reference voltages required for multilevel sensing (Figure 2.12).

The timing control for this scheme is shown in Figure 2.13. In the idle state, the dummy cell voltages are connected to their respective VDC supplies through $\overline{DCP}$ and the bitlines are precharged to V_{BLP}. The cycle starts with the deactivation of the precharge devices (via EQL) and the reference devices (via $\overline{DCP}$). After sharing the cell signal charge equally with the three sub-bitlines, they are isolated from one another via the $\overline{SWT}$ switches. The dummy wordline DWL is asserted and the reference voltages from the dummy cells are placed on the three opposing sub-bitlines. The three sense amplifiers are connected to the sub-bitlines in parallel and activated. The sense amplifiers compare the three copies of the cell signal to the three references. A data bus carries the three resulting sense amplifier outputs to a buffer after which the three logic values are converted into two bits according to the function shown in Table 2-1. Column operations may occur at this point but the data that is overwritten must follow the thermometer code of Table 2-1.

TABLE 2-1. Conversion function for Furuyama's MLDRAM.

Sense amplifier Result	Two-Bit Data
000	00
001	01
011	10
111	11

Multilevel restore is performed by disconnecting the sense amplifiers from the three sub-bitlines and then reconnecting the sub-bitlines together through $\overline{SWT}$. Charge sharing will give a final voltage on the bitline equal to the correct restore value because the sub-

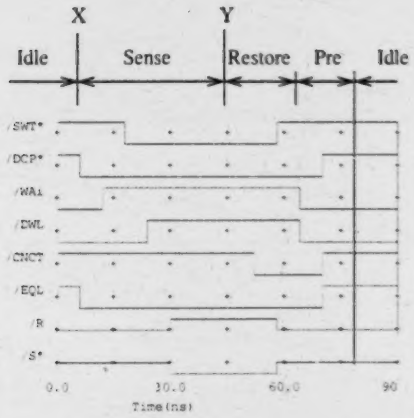


FIGURE 2.13. Timing for the Furuyama MLDRAM.

bitline capacitances are equal. The wordline is then de-asserted capturing the desired multilevel voltage back in the memory cell.

The advantages of this design are that it is fast and relatively simple. Using three sense amplifiers in parallel means that the two bits are available at the same time. However, the extra sense amplifiers imply a major area penalty that reduces the potential density gain that MLDRAM technology offers. The main disadvantage of the scheme is that it is susceptible to sensing errors due to improper reference values. The potential for such errors arises from the use of global reference voltages (VDCA, VDCB, VDCC) that need to be generated on chip and distributed across the array. A slight inaccuracy in the global reference voltage levels would be enough to cause sensing errors.

2.6.3. Gillingham's MLDRAM

Another MLDRAM scheme is proposed by Gillingham [1]. A charge-sharing restore technique similar to that in [7] is employed, but the sensing method is quite different. The architecture consists of cell columns implemented using two pairs of sub-bitlines, with each pair having a sense amplifier at one end (Figure 2.14). The four sub-bitlines can be connected in six different ways using a transistor switch matrix controlled by signals EL, $\overline{C}$, X, C, $\overline{X}$, and ER. As well, each sense amplifier can be disconnected from its respective

sub-bitline pair by de-asserting signals IL and IR. Unlike the parallel operation of the sense amplifiers used in [7], this scheme uses sequential sensing. The reference level used in the second sensing operation comes from the result of the first sensing operation. Referring to Figure 2.14, an initial MSB sensing operation compares the multilevel data to $V_{BLP}=\frac{1}{2}V_{DD}$ (the midpoint reference voltage from Figure 2.1 on page 11). The reference for the final sensing operation is generated by storing the sensed MSB value (either V_{DD} or V_{SS}) back into the accessed cell, precharging three sub-bitlines to $\frac{1}{2}V_{DD}$, and then sharing the MSB charge from the cell onto these same three sub-bitlines. Meanwhile, the fourth sub-bitline holds a second copy of the cell signal. If the first sensing operation reveals that the cell voltage is above $\frac{1}{2}V_{DD}$ then the second sensing operation will compare (after dilution) the second copy of the cell voltage to an LSB reference of $\frac{5}{6}V_{DD}$. Conversely, if the first sensing operation reveals that the cell voltage is below $\frac{1}{2}V_{DD}$ then $\frac{1}{6}V_{DD}$ is chosen as the LSB reference. The results of the first and second sensing operations produce the MSB and LSB values, respectively, for the one addressed cell. At this point the data is latched at the sense amplifiers and is ready for reading and writing.

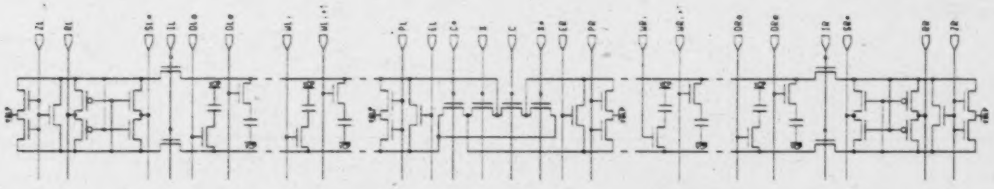


FIGURE 2.14. Schematic for the Gillingham MLDRAM.

As in [7], the techniques used in Gillingham's MLDRAM involve exploiting the carefully matched sub-bitline capacitances. An added feature of this design is the use of dummy cells. Unlike Furuyama's scheme, the dummy cells are not used for reference storage. They are used instead to cancel out the charge injection effects of wordline switching. In the circuit operation the dummy wordline is de-asserted just before a

wordline is asserted and it is asserted just after the wordline is de-asserted. In this way the noise injected across the parasitic gate-to-drain capacitances of the cell access transistors will be cancelled out.

The circuit timing is shown in Figure 2.15. In the idle state the sub-bitlines are precharged and isolated from each other and the sense amplifiers. The sub-bitline precharge devices are switched off (by de-asserting EL, ER, PL and PR) just before a wordline is activated. Consider wordline WLi. The cell voltage is shared across the two lower sub-bitlines by pulsing signal $\overline{C}$. The cell signal, having been diluted by two sub-bitline capacitances, is captured on each of the two sub-bitlines. The right lower sub-bitline remains isolated and holds the cell signal for LSB sensing while the left lower sub-bitline is used to create the reference. The left sense amplifier is connected to the left sub-bitlines through IL and the diluted cell signal voltage is compared to $\frac{1}{2}V_{DD}$. The result of this sensing operation is the value of the MSB. The full-swing MSB signal is captured in the memory cell by deactivating WLi. Then the left sub-bitlines are isolated, equalized and precharged to prepare for LSB reference generation.

The LSB reference is generated by dumping the MSB signal (either V_{DD} or 0) onto three sub-bitlines. Both left sub-bitlines and the right upper sub-bitline are shorted together by asserting EL and C, and then the MSB is dumped from the cell back onto the three sub-bitlines. With the LSB reference in place, along with the second copy of the cell signal, the LSB is sensed using the right sense amplifier. At this point both the MSB and LSB are available for reading and writing via the data bus.

The multilevel restore is performed by charging two sub-bitlines to the value of the MSB and one sub-bitline to the value of the LSB. The right sense amplifier is deactivated and the $\overline{X}$ signal is asserted causing one right sub-bitline to be charged by the left sense amplifier to the value of the MSB. The other right sub-bitline is isolated and holds the LSB voltage. When ER is asserted the three sub-bitlines (two with the MSB voltage and one with th e LSB voltage) are connected together and the common signal settles at the correct

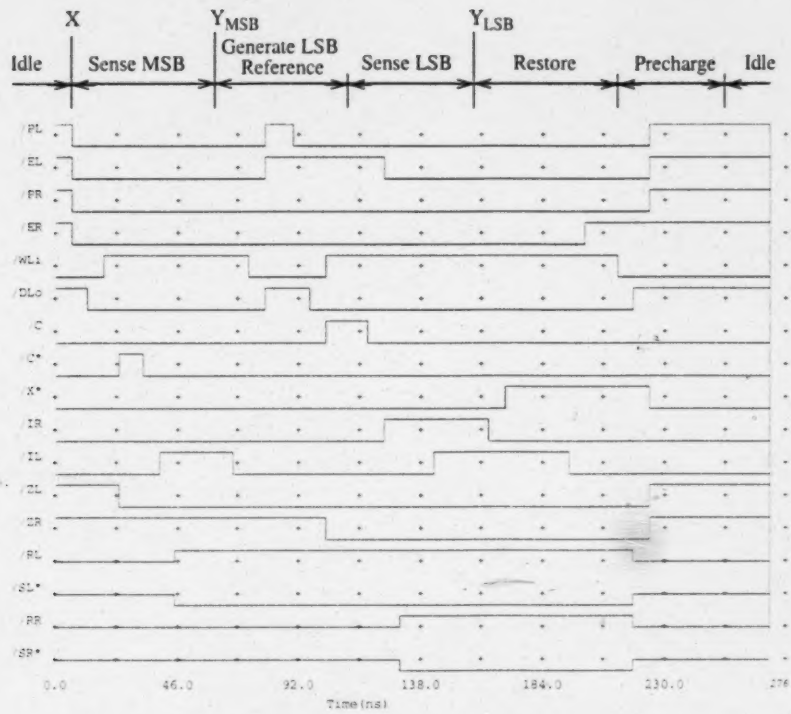


FIGURE 2.15. Timing for the Gillingham MLDRAM.

restore voltage. The wordline is de-asserted capturing the data in the cell. The bitlines are then precharged and the circuit re-enters the idle state.

The major advantage of [1] compared to previous MLDRAM schemes is that this scheme uses local components to generate the reference voltages, rather than special reference cells and power supplies as in [7]. Thus, any gradual parametric differences across the chip will not affect the reference quality. All of this should lead to higher yield and greater reliability. Another redeeming quality is that the circuit is composed of proven standard DRAM sub-circuits. There are no special capacitors or cells. Thus the departure from proven techniques is limited to the operational and configuration of proven core circuits rather than the optimized cell array.

A drawback to [1] is the extra time required for two-step sequential sensing. There are also some required operations involving the bitline charging between sensing and restoring that add time to the cycle. Although there are only two sense amplifiers per column, significant area is taken by the central switch matrix area so that the area reduction over Furuyama's scheme may be insignificant. Another drawback is the complexity of the control logic required. In particular, the scheme calls for multiple transitions on the wordline and the dummy wordline (used for noise cancellation and capacitive balance). Multiple transitions may be difficult to generate consistently in production parts. The dummy wordlines may require additional drivers and charge storage areas for wordline boosting.

2.6.4. Okuda's MLDRAM

The third MLDRAM scheme that we are considering was proposed by T. Okuda *et al.* [8]. For this experiment we are only interested in the multilevel sense and restore method so the bitline time-multiplexing and offset cancellation aspects of the design were left out. These techniques have been demonstrated [9] to be an effective method of increasing device reliability, and could be incorporated into any of the MLDRAM schemes considered.

As in [1], the cell data is extracted using sequential sensing. However, rather than using local charge sharing to generate the LSB reference voltages, a capacitive coupling method is used. The value of the most significant bit is placed onto a capacitor which bumps the second reference value up or down from the middle reference to one of the two possible LSB references. The restore scheme is similar to the previous two, but the architectural configuration is not inherently balanced so great care is required to ratio the bitline capacitances properly.

The circuit configuration, shown in Figure 2.16, has two sense amplifiers and a simple capacitor cross-connect circuit. The sub-bitlines may be connected left-to-right through the signal TG. When TG is off, the sub-bitlines are connected diagonally through coupling capacitors. Unlike the previous two schemes, where the bitlines are divided into equal sub-

bitlines, this scheme calls for unequal bitline division such that one side of the cross-connect has twice the bitline capacitance as the other.

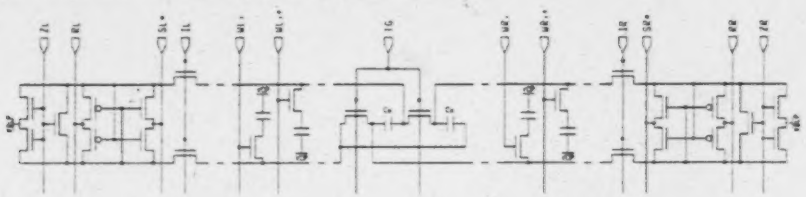


FIGURE 2.16. Schematic for the Okuda MLDRAM.

Consider the case where the left sub-bitline has the higher capacitance, and consider an access to WLi. Referring to Figure 2.17, the starting state for the circuit has the bitlines precharged, equalized and connected together through TG. First, the sense amplifiers are disconnected and the wordline WLi is asserted causing the cell voltage to be shared across the left and right sub-bitlines. The sub-bitlines are isolated from one another and the larger of the two, the left in this case, is connected to its sense amplifier. The left sense amplifier compares the diluted cell signal to $\frac{1}{2}V_{DD}$. The sense amplifier drives either V_{DD} or 0 onto the sub-bitline. This causes a $\frac{1}{2}V_{DD}$ voltage change on one sub-bitline and an equal and opposite voltage change on the other through the coupling capacitors. Careful selection of the coupling capacitor Cx has the effect of adding or removing just enough charge from the right sub-bitlines as to create the correct bitline signal for LSB sensing. The right sense amplifier is activated and the two bits become available for reading and writing.

Charge sharing is used for restore in the same fashion as the Gillingham scheme. The left sub-bitline is charged to the MSB value and the right sub-bitline is charged to the LSB value. After the sub-bitlines are isolated from the sense amplifiers, they are shorted together through TG and the correct restore value is achieved. The wordline is deactivated and the data is captured in the cell. Following restore the bitlines are returned to their original precharge state.

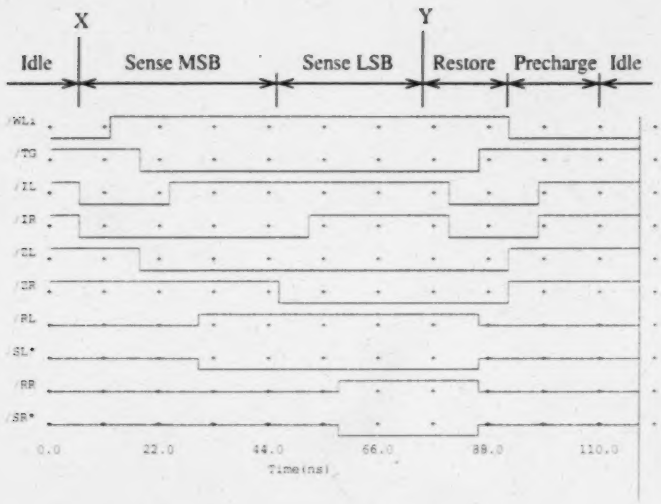


FIGURE 2.17. Timing for Okuda MLDRAM.

The main advantage to this scheme is its simplicity. The extra circuitry needed is minimal and the control timing is relatively simple. The most vulnerable aspect of this design is the use of the coupling capacitors to produce the LSB reference. These capacitors must be exactly proportioned to the cell capacitance since any error in the ratio translates directly into a reduction in the noise margins. This makes the design susceptible to processing deviations. As well, there may be some error associated with the uneven partitioning of the bitline into sub-bitlines that are related by a parasitic rather than geometrical symmetry.

2.7 MLDRAM Challenges

Practical MLDRAM designs must overcome many challenges. The main inherent problem with dividing the voltage range into more than two levels is that the noise margins between those levels decreases drastically. This is a consequence of the fact that differential voltages presented to the sense amplifiers are drastically reduced. For example, in a four level MLDRAM the noise margins are reduced by a factor of three. Overcoming this intrinsic MLDRAM problem is the key to MLDRAM circuit design, regardless of what particular scheme is used. Thus, an economically viable MLDRAM

scheme must not only have the capacity to sense and restore multiple voltage levels, it must also be robust enough to overcome any problems that reduced noise margins may cause. The second major difficulty in MLDRAM is the likely increased time it takes to perform multilevel sense and restore. The added circuitry complexity, whatever it may be, implies that the circuit timing is increasingly complex and thus likely to require more time. However, a slow MLDRAM is still an MLDRAM whereas an MLDRAM that cannot overcome noise margin problems is not useful at all.

Some of the problems in MLDRAM that can be ignored or that do not exist in DRAM that come about because of reduced noise margins are cell leakage, which reduces retention time, soft errors due to alpha particle radiation, charge injection effects caused by wordline switching, and erroneous sensing due to inherent input offset in the sense amplifiers.

Provided that the effects of reduced noise margins can be dealt with, the method of MLDRAM sense and restore can itself introduce problems. For example, in a two-bit per cell MLDRAM, there is only a potential capacity gain of two, so the sense and restore circuitry should not take up so much area that this gain is lost. The overall die efficiency should boast a significant improvement over DRAM, otherwise the MLDRAM scheme will not be worth the effort. Other problems that can be introduced by the design are parameter sensitivities and environmental sensitivities. A scheme that depends on an exact capacitance match between two structurally different capacitors may fail in production because of inherent variations in processing dimensions and electrical parameters. Similarly, a design that depends heavily on exact globally generated and distributed voltages may fail over the range of normal chip operation.

CHAPTER 3: DESIGN OF AN MLDRAM TEST CHIP

3.1 Design Overview

The goal of the test chip is to demonstrate the principles of the sensing scheme presented by Gillingham [1], the only MLDRAM from Chapter 2 that had not yet been implemented. The design is based on MOSAID's HDRAM macrocell for the Taiwan Semiconductor Manufacturing Corporation's (TSMC) 0.35μm process offered by the Canadian Microelectronics Corporation (CMC). The design strategy was to modify the HDRAM core so that it contains the required circuit structures for MLDRAM functionality.

The HDRAM core provided by MOSAID was modified to include 8 bitline pairs and 128 wordlines for a total of 1024 memory cells. Full row and column decoding was provided. The control path was omitted to provide greater flexibility to explore aspects of the control timing.

The floorplan and block diagram of the chip are shown in Figure 3.1 and Figure 3.2, respectively. The diagrams describe the physical layout and basic connections between the basic DRAM building blocks. The chip pad signals are also shown in Figure 3.2 to illustrate how they interact with the design blocks. As well, the cross connect, a switch matrix needed for multilevel DRAM sense and restore, is included.

3.2 Design Details

3.2.1. The DRAM Core

Balancing design time and the need for a reasonable core size led to the selection of a 1024-cell core. To achieve the required multilevel DRAM configuration, two major changes were made to the core. First, another set of sense amplifiers was added to each bitline pair. Second, each folded bitline-pair was split in the center and a set of eight switches, known together as the cross-connect, was added between each half. Each half bitline-pair is called a sub-bitline-pair and accommodates 64 wordlines. Because of the

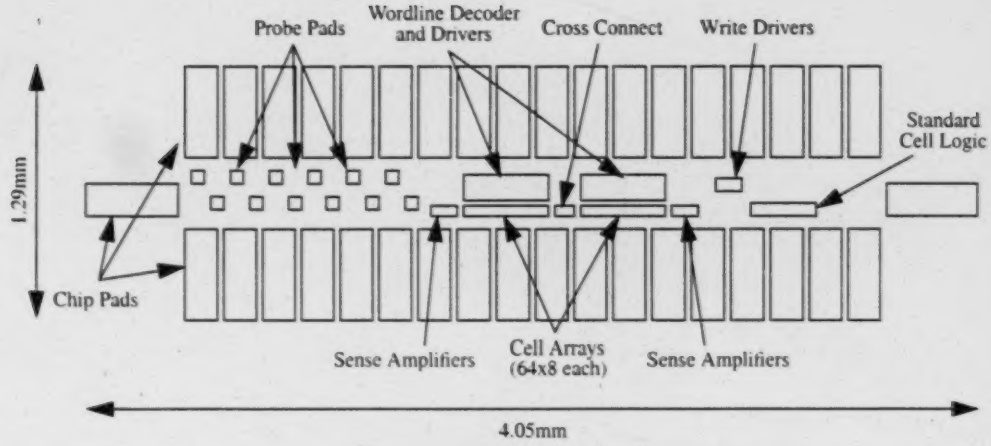


FIGURE 3.1. MLDRAM chip floorplan.

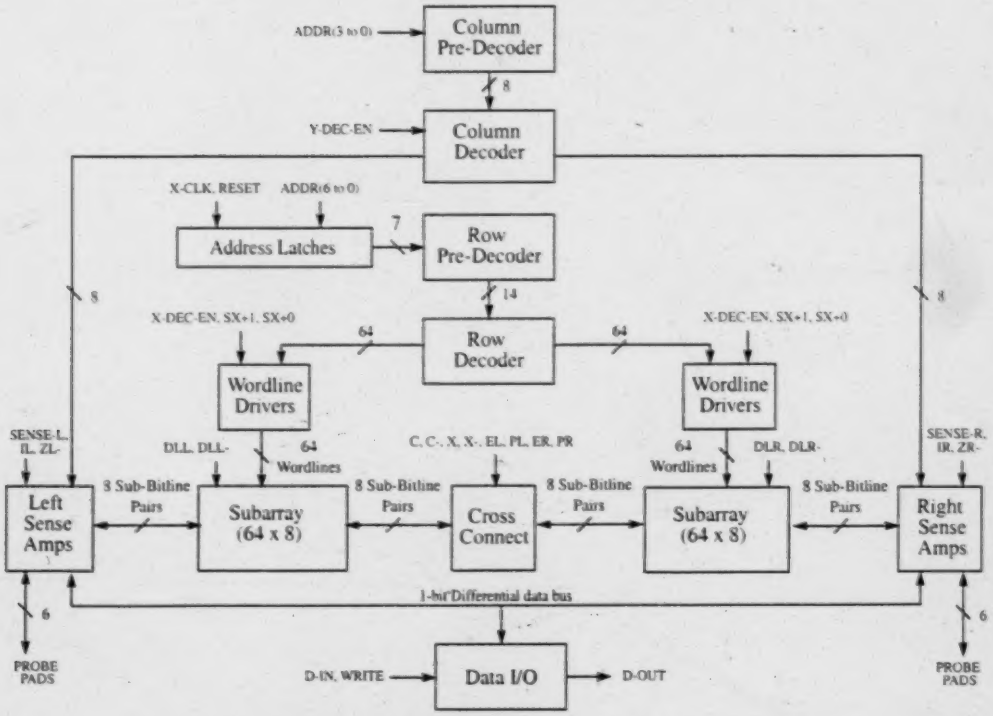


FIGURE 3.2. Block diagram of the MLDRAM test chip.

symmetry about the cross-connect, half of the core was designed and then copied, mirroring it about the central cross-connect area.

3.2.1.1 Sense Amplifiers

Since MLDRAM noise margins are reduced by one third, it was suspected that the sense amplifiers supplied in the HDRAM design might be susceptible to sensing errors from inherent input offsets voltages. We explored an offset cancellation method based on the design presented by Kawahara *et al.* [9]. The method involves using a pre-amplifier that boosts the small signal while simultaneously rejecting the effects of any improperly matched transistors within the pre-amplifier circuitry. The boosted signal is then presented to a normal sense amplifier. Should there be any inherent offset in the sense amplifier, its effects would not be manifested as a sensing error because the signal would be pre-amplified to a higher voltage than the expected inherent offset voltage.

A simulation study was performed to determine weather this offset cancellation would be necessary. The goal of the simulation was to determine how much variation is required before a sensing error occurs. Normal HSPICE simulation does not allow direct measurements of abstract properties such as sense amplifier threshold voltage, so an algorithm was designed to search for the sense amplifier threshold voltage by running multiple simulations, each of which uses a different signal value for a given mismatch. The signal value was set as an initial condition to the HSPICE simulation and the simulation was performed many times until the sense amplifier threshold was crossed.

A Perl script was written to execute the simulations over the desired range of initial signal voltages (Appendix A). Once the threshold is crossed, the script uses a binary search algorithm to obtain a user-specified degree of precision of the inherent offset voltage. Simulations were run for a single MLDRAM bitline pair to find the three sensing thresholds over a range of channel lengths for one of the sense amplifier transistors. In the experiment the normal channel length was 0.7 µm. Two sets of simulations were performed. The set of simulations had one of the two NMOS device lengths varied from 0.35 µm to 1.05 µm and the second set had one of the PMOS device lengths varied across the same range. The results (Figure 3.3) show that the inherent sensing voltage, referred back to the cell voltage, varies with the imposed imbalances in the sense amplifier. The following observations can be made from the graph in Figure 3.3. First, across the range of

lengths presented, there will never be an error in sensing a perfect cell signal of 0 V (00) or 3.3 V (11). Second, a sensing error may occur for a perfect cell signal of 1.1 V (10) if any one device in the sense amplifier is 0.35 μm while the others remain at 0.7 μm. Third, a sensing error may occur for a perfect cell voltage of 2.2 V (10) if one of the sense amplifier devices is around 0.4 μm while the others remain at 0.7 μm.

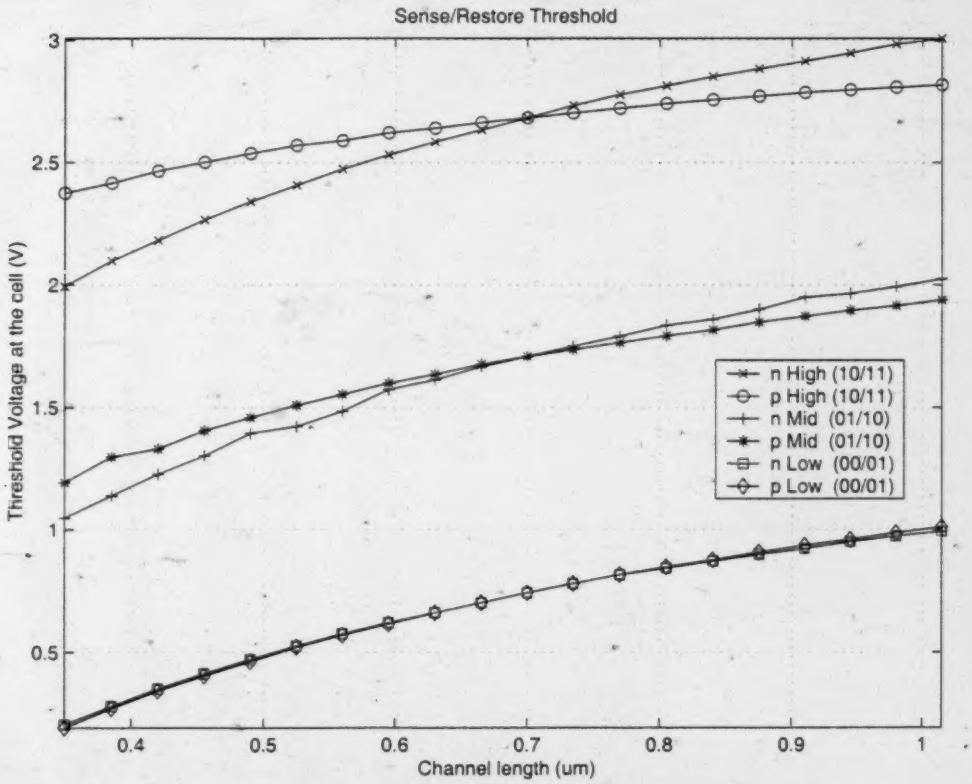


FIGURE 3.3. Reference thresholds in sensing referred to the memory cell as a function of channel length in the sense amplifier. Legend text "n High" refers to the threshold between storage levels 10 and 11 when one of the NMOS channel lengths is varied.

Based on the above observations, it seems unlikely that a sensing error would occur due solely to a sense amplifier mismatch of the type described. If the cell signals are degraded from the ideal, however, the channel width mismatch needed for a sensing error decreases, as can be seen in Figure 3.3. However, it seems offset cancellation is not needed

because the expected channel length mismatch between adjacent transistors is expected to be much less that one percent. Thus, offset cancellation was not incorporated into this design. Although there are other variables that may cause inherent offset, after viewing these results one may guess that the other effects will fit into similar curves and so the overall robustness on the given HDRAM sense amplifier is suitable and does not require modification for use in this test chip.

The given HDRAM sense amplifiers already fit in the pitch of one bitline pair so no extra layout work was needed to fit two sense amplifiers in the bitline pitch (one at either end). The only functional change to the sense amplifiers was the addition of isolation devices between the sense amplifier nodes and the bitlines. These are required for the many charge sharing operations employed in the multilevel sense and restore cycle. The placement of the sense amplifiers with respect to the bitlines and isolation devices is shown in Figure 3.6 below.

Of the sixteen sense amplifiers in the design, two were specially modified further to include test points for observing the analog behavior of one full bitline. These special sense amplifiers are on the second last bitline. The supporting circuitry for these test points is shown in Figure 3.4. This circuit has the NMOS transistors gates connected to the bitlines and the drains connected together. The TP0, TP1 and TPCOM signals are available at the chip probe pads. With the appropriate external circuitry, amplifiers can be created to compare differential bitline signals at different times and locations along the folded bitline. Example experimental setups to observe the differential bitline signals are shown in Figure 3.5. For the common source configuration, the load resistors can be chosen to give a desired gain while the source follower can be designed for unity gain.

In total there are 4 pairs of test points. The specific locations of the test points are shown in Figure 3.6. There it can be seen that, for the one bitline that has this functionality, there is one test point on each sub-bitline, and one on each of the two sense amplifier nodes. They are controlled by a total of 12 probe pads from which 4 differential sense amplifiers can be constructed.

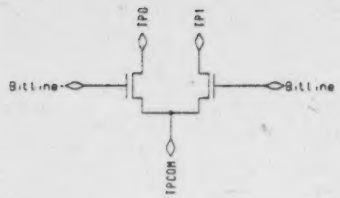
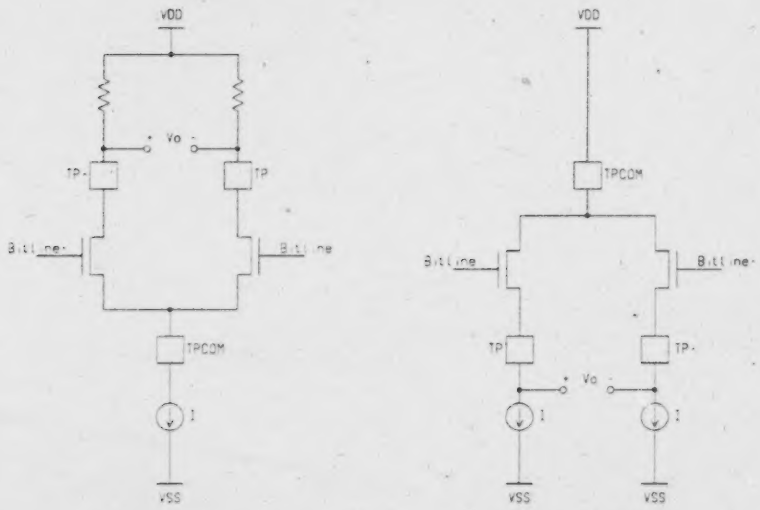


FIGURE 3.4. Test point circuitry.



Common Source Source Follower

FIGURE 3.5. Example external circuitry for probing the bitlines.

3.2.1.2 Bitline Twists and Dummy Cells

HDRAM features three twist regions in the cell array for noise cancellation. These twists help to reduce noise that is induced via stray capacitance between adjacent bitlines. To keep the same level of noise cancellation in multilevel DRAM as in HDRAM, three twist regions were added to each sub-bitline. Although this added more overhead to the array, increased die efficiency was far less important a goal for our experiment than was increased noise immunity.

Another feature added to the array are the dummy cells. The multilevel scheme under test requires four dummy cells per bitline pair, one for each sub-bitline. These cells were

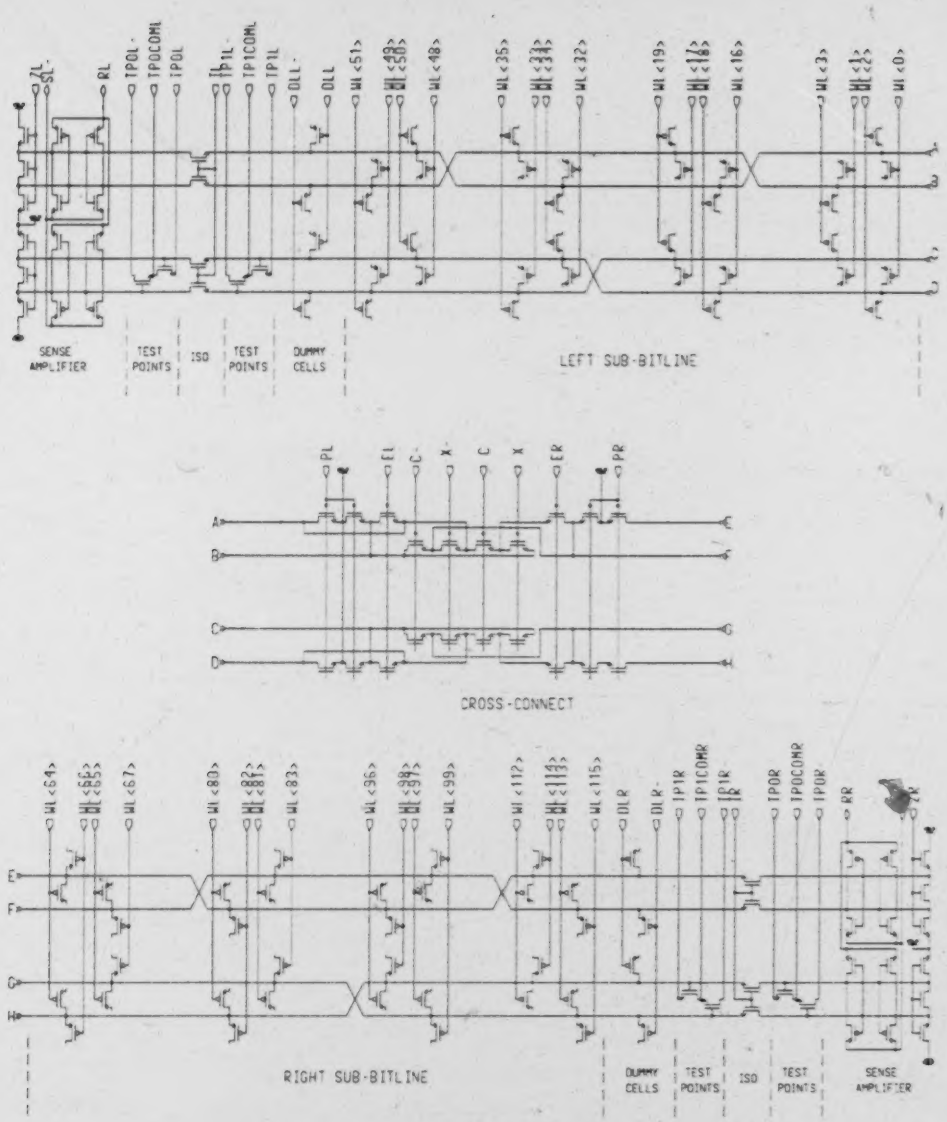


FIGURE 3.6. Arrangement of the core components.

added near the sense amplifiers. Each dummy wordline is associated with either an odd or an even wordline address. When twists were added to the bitlines, a slightly more complicated wordline arrangement was necessary to preserve the odd/even association. The wordline addressing and the arrangement of the connections between cell and bitline

is shown in Figure 3.6. Between each twist of the bitlines there is a change in the cell-to-bitline connection to keep each even wordline connected to a cell on the same bitline as DLL (DLR-).

3.2.1.3 Wordline Drivers

Wordline boosting for PMOS cell access transistors is a challenge since the wordlines must be driven below V_{SS} (by about -2 V) to allow a full V_{SS} value to be written into the cell. This prevents the use of n-channel transistors for wordline drivers because driving the source of an NMOS transistor below V_{SS} would create a forward-biased p-n junction to the substrate. Instead, HDRAM uses a self-bootstrapped PMOS driver (Figure 3.7). In HDRAM, a charge pump is used to maintain a negative voltage on an on-chip capacitor. This pump is then used to deliver the negative voltage signal to the wordlines. HDRAM employes a large capacitance to store enough charge to drive the 2000+ transistor gate loads of the wordline. In adapting the design for this multilevel DRAM test chip, the charge pump scheme was abandoned altogether. Instead, the basic self-bootstrapped wordline driver is operated through off-chip control. The terminals of the charge pump capacitors were grounded and the charge pump control circuitry was removed.

The timing for wordline driver is shown in Figure 3.7. A wordline is selected via the SELECT signal, which comes directly from the row decoder. The selected wordline is activated by a rising transition on X-DEC-EN followed by a falling transition on SX+ from a high voltage (V_{DD}) to the negative boosting voltage. The rising transition on X-DEC-EN causes the source of the transistor M1 to go from V_{DD} to V_{SS}. As this voltage drops, the M1 drain voltage follows the source voltage until the source node falls below one PMOS threshold above V_{SS}. At that point, M1 turns off. The drain of M1, and hence the gate of the PMOS drive transistor (M2), becomes isolated and floats at one PMOS threshold voltage above V_{SS}. When the falling transition occurs on SX+, the parasitic capacitance C_{GS} in M2 causes the floating gate of M2 to be bumped well below the boost voltage, thus turning M2 on and allowing the wordline to be activated. While activated, the

wordline may be switched off at any time by de-asserting X-DEC-EN. This disconnects the wordline from SX+ and causes M3 to turn on and restore the wordline voltage to V_{DD}.

The dummy wordlines do not use this structure. Rather, each dummy wordline is directly connected to a pad to give full control over dummy wordline operation. The onus is on the tester to provide this signal at the boosted voltage.

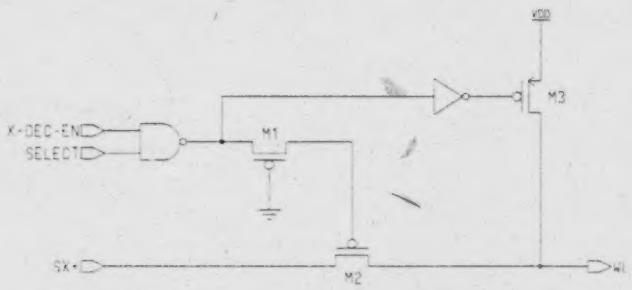


FIGURE 3.7. Wordline driver schematic

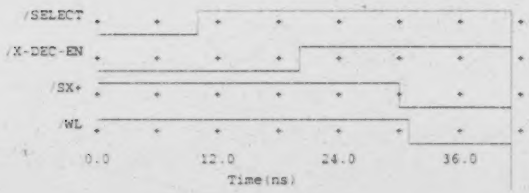


FIGURE 3.7. Example timing for turning a wordline on

The row decoder is also included in the HDRAM wordline driver circuitry. The HDRAM row decoder provides 7-bit decoding and a single decode enable input. The decoder is arranged so that the six most significant address bits select one pair of wordlines. The address LSB then selects which wordline of the pair is to be activated. Thus, the wordlines are numbered 0, 2, 1, 3,... (in Figure 3.6 above).

3.2.1.4 Row and Column Address Scrambling

The address scrambling function is depicted in Figure 3.8. The left sense amplifiers are addressed 0 through 7 while the right sense amplifiers are addressed 8 through 15. The

rows begin with 0 in the left array closest to the cross-connect circuitry and follows the pattern shown.

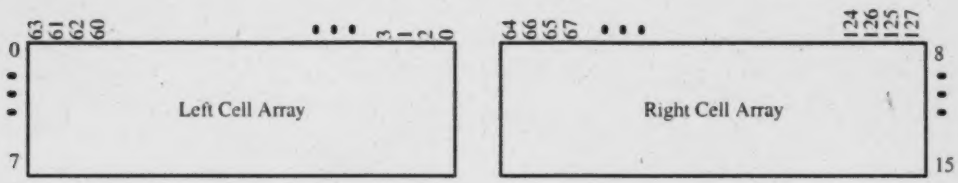


FIGURE 3.8. De-scrambled row and column address map

3.2.2. The Peripheral Circuitry

The periphery is made up of a row and a column pre-decoder, a column decoder, address latches, data I/O, probe pads and bonding pads. All of these blocks were implemented with standard cells and connected using automatic place-and-route tools except for the data bus drivers, the data bus precharge switches, and the probe pads.

3.2.2.1 Row and Column Pre-decoders

The row pre-decoder uses seven flip-flops to hold the row address. These flip-flops are positive edge sensitive to a latch strobe pin. Once latched the row address is pre-decoded and presented to the row decoder. The final wordline is activated by a global row-decode-enable pin (X-DEC-EN in Figure 3.7) and the subsequent voltage change on the wordline boosting signal (SX+) from V_{DD} to the negative boost voltage.

Unlike the row pre-decoder, the column pre-decoder does not latch the data. Instead, the pre-decoder and decoder are implemented in one block of combinational logic. The decoded address of the first four address pins is continuously decoded. Final control over the column access is given to a column-decode-enable pin. When asserted, one of the sixteen sense amplifiers is connected to the data bus.

3.2.2.2 Data I/O

A differential data bus connects all sixteen sense amplifiers to the I/O logic. Due to pad limitations, the data bus is only one bit wide. To obtain the sign and magnitude bits for

a multilevel read, two consecutive sensing operations must be performed. The data bus remains precharged until the column-decode-enable signal is asserted, causing the precharge devices to turn off. The read data is always present at the output pin but is only valid at the appropriate time during a read operation (a column access).

The data I/O circuitry is shown in Figure 3.9. The output driver is always enabled so the value of the data bus is continually available at the DATA-OUT pin. The value on this pin is not valid until the Y-DEC-EN signal is brought high, causing the data bus to be disconnected from the precharge voltage V_{BLP}. The Y-DEC-EN signal also enables the row decoder to access one of the sixteen sense amplifiers, thus the databus becomes charged with the data on the selected sense amplifier. The write operation is done through large data bus drivers that can, if necessary, flip the state of an accessed sense amplifier. The data is placed at the DATA-IN pin and the WRITE pin is brought high. It is crucial that the column decoder be enabled whenever writing. Otherwise, there will be contention between the write drivers and the data bus precharge.

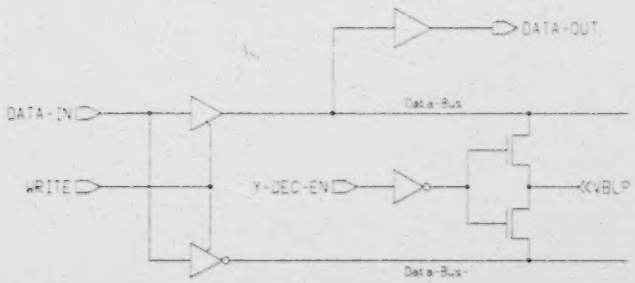


FIGURE 3.9. Schematic of the Data I/O circuitry.

3.2.2.3 Chip Pads

The chip power supplies are V_{DD}, V_{SS}, V_{BLP} (bitline precharge) and V_{CP} (cell plate bias). Standard pads were used for V_{DD} and V_{SS}. For V_{BLP}, a standard low impedance analog input pad was used. A custom pad for V_{CP} needed to be made because V_{CP} is to be used at least one threshold above V_{DD}. For the custom pad the ESD device and V_{DD} clamping diodes were removed to allow the high voltage to be passed. This pad was also

used for control signals that need to be boosted above V_{DD}. These boosted signals are IL and IR (isolation), and EL, ER, X and X- (cross connect). Custom pads were also created for the wordline boosting signals SX+1 and SX+0 and the dummy wordline signals, all of which require voltages well below V_{SS}. For these, all ESD protection and all diodes were removed to allow any voltage to be passed into the chip. Table 3-1 lists all of the pads as well as the pad type. The probe pads were custom made and provided with an opening of 64μm by 64μm in the passivation.

TABLE 3-1. Test Chip Pads

Pad Name	Type	Pad Name	Type	Pad Name	Type	Pad Name	Type
RESET	Digital I/P	DLL	Custom	PL	Digital I/P	IR	Analog I/P
VCP	Custom	DLL-	Custom	X	Analog I/P	SENSE-R	Digital I/P
VBLP	Analog I/P	DLR	Custom	X-	Analog I/P	D-IN	Digital I/P
ADDR(6 to 0)	Digital I/P	DLR-	Custom	C	Analog I/P	Y-DEC-EN	Digital I/P
X-CLK	Digital I/P	ZL-	Digital I/P	C-	Analog I/P	WRITE	Digital I/P
X-DEC-EN	Digital I/P	IL	Rnalog I/	PR	Digital I/P	D-OUT	Digital O/P
SX+1	Custom	SENSE-L	Digital I/P	ER	Analog I/P	VDD	Power
SX+0	Custom	EL	Rnalog I/	ZR-	Digital I/P	VSS	Ground

When powering up the chip, the proper voltages must be present on the power pins. Since this is a test chip there is no protection against improper input combinations that can cause high currents within the chip. For example, asserting the SENSE-L signal while ZL- is asserted will cause a short circuit between V_{DD} and V_{BLP} and between V_{SS} and V_{BLP}. Table 2 shows a safe power up state where all bitlines are precharged and all sense amplifiers are de-activated.

TABLE 3-2. Power-up State

Signal	Value	Signal	Value
VDD	3.3V	IL, IR	3.3V
VSS	0V	ZL-, ZR-	0V
VCP	4.5V	PL, PR	3.3V
VBLP	1.65V	Y-DEC-EN	0V
SENSE-L	0V	WRITE	0V

CHAPTER 4: TEST RESULTS

Two different chips were fabricated from the chip design presented in Chapter 3. Due to a place-and-route error, the first chip was not usable as an MLDRAM, but it was tested as a 1-bit/cell DRAM. The second chip was tested both as a DRAM and an MLDRAM. This chapter first describes the testing of the first chip in DRAM mode, and then describes the DRAM and MLDRAM mode testing of the second chip.

4.1 Test Equipment

The MLDRAM chips were tested using an IMS LogicMaster XL tester. The tester, intended for testing digital chips, provided enough functionality to test the MLDRAM chips. It has 64 digital I/O channels each with the capability of switching between user defined high and low voltages (V_{HIGH} and V_{LOW}). These two voltages can be independently chosen for each pin. Test vectors can be applied to the device under test (DUT) on every tester clock, which has a minimum period of 16.675 ns. In one mode of testing, pin signals are changed only at the start of clock periods. This mode uses a signal format called non-return-to-zero (NRZ). In an alternate mode of testing, each channel can be programmed to deliver a pulse with a programmable delay and programmable pulse width. The delay and pulse width can be specified to a resolution of 100 ps thus allowing the complex timing of the chip to be realized within two tester clock periods. The tester format, called return-to-zero (RZ), causes a pulse with a specified duration to occur within a tester clock period when the vector value is 1 and causes no change in the output (i.e. it remains at V_{LOW}) when the vector value is 0.

4.2 Test of the ASIC DRAM Chip

After the chip was manufactured, I discovered missing power and control wires for the left set of sense amplifiers. This error, introduced by the Cadence place-and-route tool, rendered the left array unusable and so made MLDRAM operation impossible. Nevertheless, the chip was tested as a 1-bit/cell DRAM by using only the right memory

array and its associated sense amplifiers. The following sections describe how the chip was mounted and tested using the IMS tester. The results of DRAM testing show that it functioned properly as a 1-bit/cell DRAM.

4.2.1. DUT Board Design

The five prototype chips were packaged in 64-pin ceramic PGA packages. Unfortunately, there was no PGA interface card immediately available. I had available, however, a 64-pin DIP socket board. Therefore, an adapter board was required to connect the PGA package to the DIP interface.

I constructed the adapter board using wire-wrap technology. We purchased a PGA zero-insertion force (ZIF) socket and mounted it on the wire-wrap breadboard. The chip pins used only for MLDRAM operation and those affected by the place and route error were wired directly to the power supplies. The others, needed for operating the right array in DRAM mode, were wired so that each was connected to one of the DIP pins in the tester's socket card. The boosted wordline controls, the precharge and the cell plate pins of the chip were also connected to tester outputs because the tester provides the flexibility to specify independent voltages for each pin. To prevent noise, the adapter board was equipped with decoupling capacitors for V_{DD}, V_{BLP}, and V_{CP}.

4.2.2. Test Vector Generation

Two files are required to use the IMS tester: a setup file and a vector file. In the setup file, the chip inputs and outputs are declared. For ease of testing, each input was set to change on the IMS tester's internal clock edge. Thus, for a given output, the vector value of 1 causes the tester output to be forced to V_{HIGH} for the duration of the clock period, and vice-versa. With this setup, the timing control was coded into vectors so that for each timing edge in the critical timing path, one vector is supplied. Using this simple setup, a C++ program (Appendix B) was used to generate vectors for various march tests. The scrambling function of the chip was incorporated into the program so that the correct physical voltages could be read and written at each stage of the march tests.

I was able to debug both the DUT board and the vector generation program. Once the DRAM functionality was verified, the timing control was compressed in time using the IMS forcing formats. These formats allow the tester outputs to be pulsed within the tester period rather than simply changing on the tester clock edge. By specifying the delay and pulse width of each tester output the timing control, shown below in Figure 4.1, I was able to realize the control timing in just two tester cycles rather than 13. In the first tester clock period, the chip is idle with the bitlines precharged and the sense amplifiers deactivated. At the same time, the row address is applied to the chip and latched into the row latches. In the second tester clock period, all of the rest of the edges shown are applied.

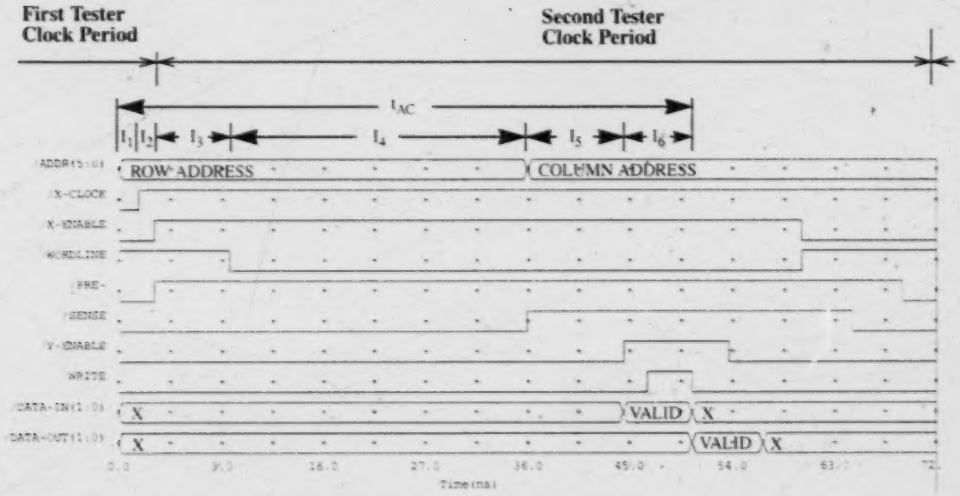


FIGURE 4.1. Control timing for the first MLDRAM chip

4.2.3. Access Time Tests

Several standard march tests were applied to evaluate the gross functionality of the DRAM arrays. All cells were functional on all five test chips under nominal conditions using a 5n march test [4]. Having functionally verified the chips, we then experimented with the timing to minimize the access time. We had the advantage of direct control over all key timing events, control that is not available to users of commodity DRAMs via the provided signals.

The strategy for finding the fastest possible timing was to consider the six key intervals within an access cycle (see Figure 4.1). Beginning with the earliest interval, each was progressively reduced while all other intervals were held at conservative values. Table 4-1 summarizes these results. An address to data-out access time of 38.7 ns was measured on all five test chips.

TABLE 4-1. Critical Timing Intervals

Interval	Bounding Edges	Value (ns)
I1	ADDR row to X-CLOCK rise	1.7
I2	X-CLOCK rise to X-ENABLE on	1.3
I3	X-ENABLE on to WORDLINE on	6.5
I4	WORDLINE on to SENSE on	14.5
I5	SENSE on to Y-ENABLE on	7.6
I6	Y-ENABLE on to D-OUT valid	7.1

Testing was performed at 3.3 V with an ambient temperature of 22 C. The memory chips function correctly down to V_{DD}=2.7 V. This result verifies that the design can function as a DRAM so, despite the place-and-route error, the basic design was successful. The second version of the chip, discussed in the next section, has the error corrected.

4.3 Test of the MLDRAM Chip

Before the first chip prototypes were returned from manufacture the design of a second version was underway. Special attention was given to the place and route step to ensure that a similar error did not occur again. Despite our best efforts, an error in the schematic database was left unchecked so two key transistors were left out of the manufactured design. However, this did not preclude MLDRAM operation and we were able to work around it by modifying the control timing and verify that the chip functions properly as both a 1-bit/cell DRAM as well as a 2-bit/cell MLDRAM.

4.4 DUT Board Design

The package for the second chip design was not a PGA but an 80-pin CQFP. Thus a new interface board was required to connect the chip to the tester PGA socket card. We decided to build a printed circuit board because we could not find a clamshell socket that would fit the package. The board connected all 40 pins of the chip (the other 40 on the package are unused) to the tester. Again, decoupling capacitors were placed on the interface board to prevent noise in the power supply pins.

4.4.1. Testing

After extensive testing of the chip in 1-bit/cell DRAM mode we discovered the error that was preventing the intended proper chip operation. Two data bus precharge transistors were absent from the chip so the databus signal retained whatever data was previously written to it. The sense amplifiers were unable to contend with opposing data bus values so we were initially unable to read any sense amplifiers.

To work around this problem, we modified the control timing so that while the bitlines are precharging between cycles, the data bus is precharged as well. This was done by leaving the column active between cycles so that it remains connected to one of the sixteen sense amplifiers during the precharge phase. This technique was successful and we were able to verify proper operation of this chip as a 1-bit/cell DRAM for all 1024 cells.

For multilevel operation, the precharging problem becomes complicated by the fact that MLDRAM requires two sequential column accesses, one for the sign bit and one for the magnitude bit. Precharging the data bus between cycles readies it for reading the sign bit, but the sign bit remains on the data bus and so causes errors in reading the magnitude bit. The remedy used for this is as follows. After the sign bit has been read at the chip pin, the column access is deactivated and the bitlines are allowed to charge back up to the voltage rails. Then, the sign sense amplifiers are disconnected from their respective sub-bitlines through the "I" (isolation) signals, leaving the sub-bitlines floating and thereby retaining the sign bit. The sign sense amplifiers are then precharged through the Z- signal and the data bus is reconnected to one of them. By this method the data bus is precharged

through the precharge devices in the sense amplifier. Immediately following this data bus precharge step, the data bus is disconnected from the sign sense amplifier and the magnitude is read out by a normal column access operation. In the meanwhile, the sign sense amplifier is disconnected from precharge and reconnected to the bitlines to rejuvenate the floating sub-bitlines. The cycle proceeds from this point along the ordinary timing control through restore and precharge to the idle state.

4.4.2. Results

The chip was first tested as a 1-bit/cell DRAM and no errors were detected. Test results show that the chip functioned properly as a MLDRAM using the tester in the relaxed timing mode (NRZ). Some failures occurred when reading the LSB but these failures were minimized by altering the wordline boost voltage to -1 V. After changing the wordline and dummy wordline boost voltage, the chip was tested across all even wordlines on the left side of the chip. Most of these cells functioned without error for the following march test.

$(w0), (r0w1), (r1w3), (r3w2), (r2w0)$

The failing cells were at the intersection of every second even wordline and every second column. Those cells that failed did so only for reading back the $\frac{1}{3} V_{DD}$ level. The possibility that an incorrect scrambling function was applied has not yet been ruled out. Also, some irregular errors appeared on rows 0 and 2.

A voltage bump test was performed to measure noise margins. Unfortunately, the IMS tester has a minimum voltage swing between V_{HIGH} and V_{LOW} of 0.6V. Thus, the minimum bump to the cell plate is 0.6V. Using this bump, 19 errors occurred from a total of a possible 128 for the 32-cell block tested. Thus, the tester lacks the functionality to do a proper voltage bump test to measure noise margins.

4.4.3. Conclusions

The MLDRAM scheme works since most of the cells functioned properly. Most of the errors occurred for a cell voltage of $\frac{1}{3} V_{DD}$, so this level has the poorest noise margin. The regular structure of the observed errors indicates that there may be a layout error or a scrambling problem in the test vectors. Further testing is needed to explain the source of these errors. The scheme is susceptible to errors due to wordline charge injection, a problem that was meant to be solved through the use of matched dummy wordline switching.

CHAPTER 5: DESCRIPTION OF AN IMPROVED MLDRAM

This chapter describes a new MLDRAM scheme that combines the speed of the MLDRAM proposed by Furuyama *et al.* and the noise cancellation techniques of the MLDRAM proposed by Gillingham. The method presented by Furuyama *et al.* [7] is fast because the activation of the sense amplifiers is simultaneous. This is possible because it uses three regular sense amplifiers to perform sensing in parallel, reminiscent of flash A/D conversion. The result is fast overall sensing time. Thus, the total random access time, from the time of application of a row address to the time of the data availability for a read operation, is minimal and comparable to the row access times of standard DRAM. The shortcoming is its dependence on internally generated global reference levels for proper sensing. The proper operation of the design is highly sensitive to the global reference levels. In reality, however, the ability to generate and distribute accurate global reference signals (denoted V_{DCA}, V_{DCB}, V_{DCC} in [7]) is problematic and unreliable over typical operating conditions and inevitable processing variations. Another drawback is the way in which the dummy cells are charged to their respective reference voltages. Furuyama uses an extra transistor (a reference switch) in the dummy cell that can connect the cell storage capacitor to a global reference voltage. The use of a reference switch within the dummy cell is difficult to build because it departs from the normal cell array structure. Also, the switch in the dummy cell introduces a small but significant overall capacitance error in the cell. The capacitance error is due to the parasitic capacitance associated with the drain of the reference switch, which has the gate controlled by the signal named DCP-. Also, the use of this switch introduces an undesirable voltage change in the cell due to charge injection from the gate of the DCP- transistor into the dummy cell.

The method presented by Gillingham [1] overcomes all dependencies on global reference voltages and has no need for dummy cell switches. The reference voltages are generated locally, for each bitline pair, at the time of sensing. This makes the design more robust and should result in more reliable operation despite variations in operating conditions and processing variations. However, the design does not use concurrent

sensing. Rather, it uses sequential sensing where the result of the first sense amplifier's sensing operation is used to generate the reference voltage for the second sensing operation. There are two disadvantages to this. First, the process of reference generation and sequential sensing is time consuming. This is generally undesirable as it directly impacts the row access time. The result is that the row access time is significantly longer than that of [7] and regular DRAM. This in itself may be enough to restrict the use of [1] to specific applications rather than serve as a replacement for regular DRAM. The second problem with this method is the use of many control switches and the resulting timing complexity. For each switch that must be activated or deactivated in the timing control, there is inevitably a small charge injected on to the bitlines. Because of the imbalance in switching between the four sub-bitlines (as defined in [1]) there is an imbalance in net charge injected into each. The result is that the signal levels presented to the sense amplifiers prior to sensing are inaccurate (see Appendix C).

It is possible to have both the fast access advantage of [7] with the local reference generation ideas of [1]. The following describes a new MLDRAM in which, like [7], each bitline pair is divided into three equal-length sub-bitline pair segments, where each segment is provided with a sense-amplifier circuit. This allows fast, single-step flash-A/D conversion sensing. Instead of using globally-generated reference voltages for sensing, the new scheme uses charge-sharing techniques between three adjacent sub-bitlines, reminiscent of [1], to locally generate the three necessary reference levels. Basically, the new MLDRAM uses the benefits and overcomes the problems of the two previous designs. It does not use global references as does [7] and so there are no switches in the dummy cells, as in [1]. It does use the parallel activation of all sense amplifiers as in [7] and avoids the complex and time-consuming reference generation and sequential sensing of [1]. As well, charge injection effects are specifically cancelled out by the use of fully symmetric and bitline-balanced control switch usage.

5.1 Detailed Description of the New MLDRAM Scheme

5.1.1. MLDRAM Circuitry

Figures 5.1 and 5.2 give schematics for the new MLDRAM. Figure 5.1 shows the two different sub-bitline pair configurations that are used and Figure 5.2 shows how nine sub-bitline pairs are organized as a 3-by-3 array. The position of each sub-bitline pair is identified by a horizontal co-ordinate {L, C, R} and a vertical co-ordinate {T, M, B}. The sub-bitlines can be connected together horizontally, L to C to R, via switches controlled by signals SWT0 and SWT1, and vertically, T to M to B, via switches controlled by signals REF0 and REF1.

The M sub-bitline pairs differ slightly from the T and B sub-bitline pairs in connections made to generate word lines GW0 and GW1 and reference wordlines RW0 and RW1. These special wordlines are identical to normal wordlines in all respects except that the cell access transistors for GW0 and GW1 are missing in the M sub-bitline pairs (following configuration SBL_RW in Figure 5.1), and the cell access transistors for RW0 and RW1 are missing in the T and B sub-bitline pairs (following configuration SBL_GW in Figure 5.1)

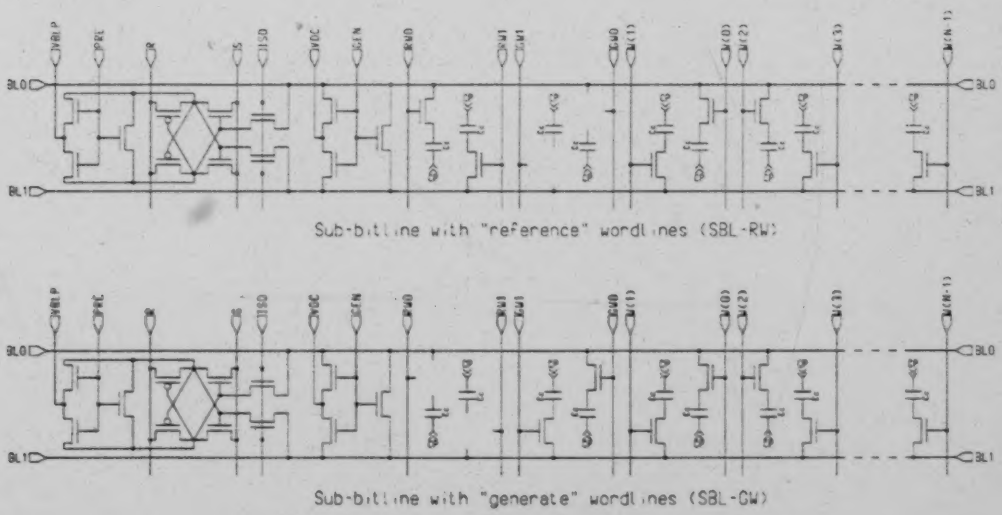


FIGURE 5.1. Sub-bitline designs.

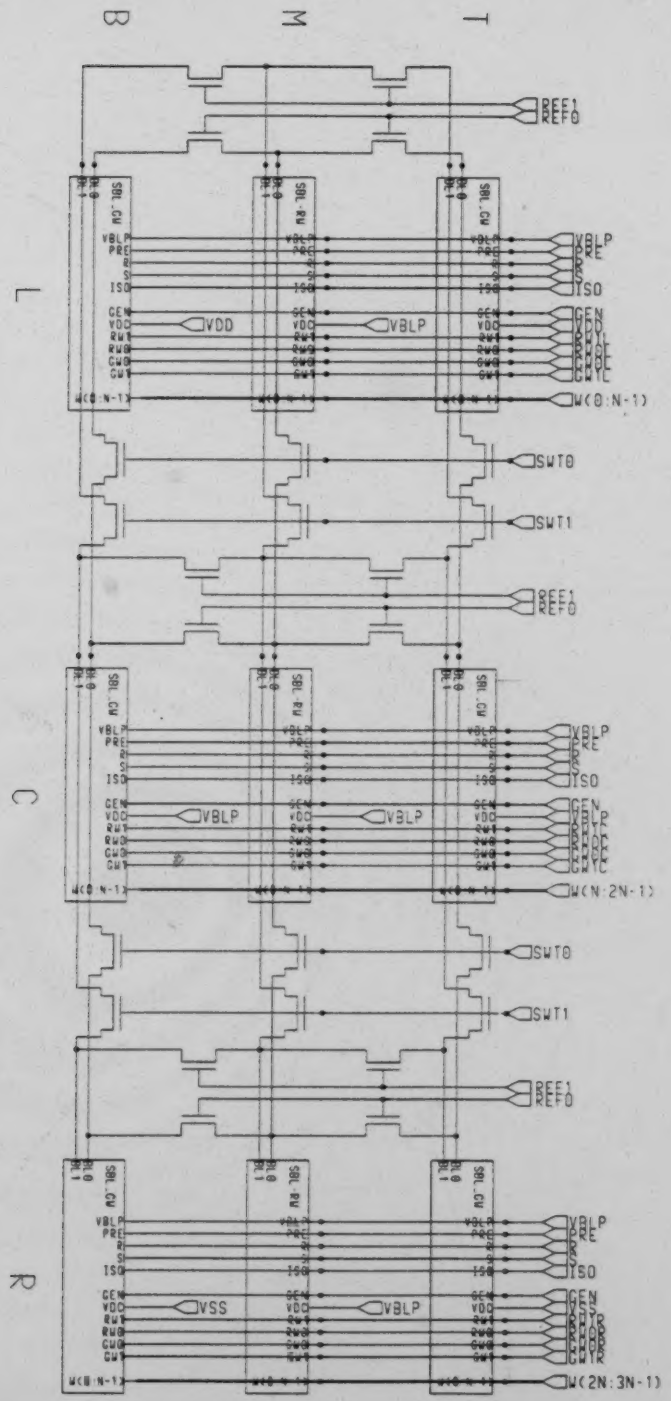


FIGURE 5.2. Top level schematic of the new MLDRAM scheme.

The sub-bitline pairs also differ in connections made to transistors controlled by the signal GEN. This signal is used to generate the reference voltages after a restore in preparation for the next sensing cycle. As shown in Figure 5.2, signal GEN connects the sub-bitlines to either V_{DD}, V_{SS} or V_{BLP}. The organization of these connections used is critical, though there are some possible variations. The signal GEN connects the sub-bitlines TL and BL to V_{DD}, TR and BR to V_{SS}, and all others to V_{BLP}.

5.1.2. MLDRAM Operation

Reference cells are provided in the ML, MC and MR sub-bitline pairs. Consider ML first. Sub-bitlines in TL, ML, and BL are precharged separately to V_{DD}, V_{BLP}, and V_{DD}, respectively, by asserting the GEN signal. To ensure that the capacitances of all sub-bitlines are equal, the RW0, RW1, GW0 and GW1 signals are all asserted causing each sub-bitline to have the same capacitance equal to the parasitic capacitance of the bitline (CB) plus one memory cell (CC). After charging, the L sub-bitlines are shorted together (T to M to B) creating the final voltage of $\frac{5}{6}V_{DD}$. The same operation is performed on the C and R groups of sub-bitlines but the charged values for these cause the resulting voltage to be $\frac{1}{2}V_{DD}$ for TC, MC, and BC and $\frac{1}{6}V_{DD}$ for TR, MR, and BR. Having created the three required reference voltages, the reference voltages are stored in the reference cells by de-asserting signals RW0 and RW1 in L, C, and R.

The operation is shown in the control timing of Figure 5.3. Up until this point, control timing waveforms have been presented starting in the idle state, followed by sensing, then restoring, and returning to the idle state. However, the control timing shown is most easily understood by starting with the restore operation and proceeding into the reference generation. After that, the circuit is idle, awaiting a random access. Thus the start of the timing control diagram has the sense amplifiers activated and the wordline asserted.

Assume that W0 (wordline-0) has just been accessed and is awaiting restore. From Figure 5.2 it can be seen that W0 falls in the left-hand (L) sub-bitlines. This is an

important observation since the location of the wordline (L, C or M) determines the operation of the reference and generate signals used for restore.

In preparation for restore, all sub-bitlines have equal capacitance because there is exactly one memory cell connected to each, and the sub-bitlines are otherwise identical. This can be seen on the control timing diagram. At t = 0 ns, the following signals are active: W0, RW1L, RW0C, RW1C, RW0R, RW1R, GW0C and GW0R. All inter-sub-bitline switches (SWTx, REFx) are off and the sense amplifiers are on. The value on the sense amplifiers represents multilevel data in a three-bit code as there are three sense amplifiers per bitline. The coding is shown in Table 5-1 and was taken from [7]. The four possible states of the three sense amplifiers are mapped to their associated binary value. Other coding schemes, such as a Grey code, may be used.

TABLE 5-1. Coding of Bits

L	C	R	Binary
0	0	0	00
0	0	1	01
0	1	1	10
1	1	1	11

As described in [7], the restore voltage is generated by shorting the three sub-bitlines together as one long bitline. The resulting voltage will be exactly as expected in the voltage code presented in [7] and [1]. The restore problem of [7] is that the sub-bitline to which the memory cell is connected represents a total capacitance greater than the other two. This unbalance causes a small error in the restore voltage. The situation is corrected in [1] by use of the dummy wordlines but the method of [1] fails to predict the error caused by inter-bitline capacitance. Between the bitline and its complement there is a small parasitic capacitance that adds to the total. In [1] the final step of restore is the activation of the "ER" signal (please see [1]). This shorts the right sub-bitline to its complement but in doing so shorts the parasitic capacitance between the two. The final result is an under-representation of the "magnitude" bit in the restore voltage. These problems do not occur in the new scheme because (a) reference and generate wordlines are

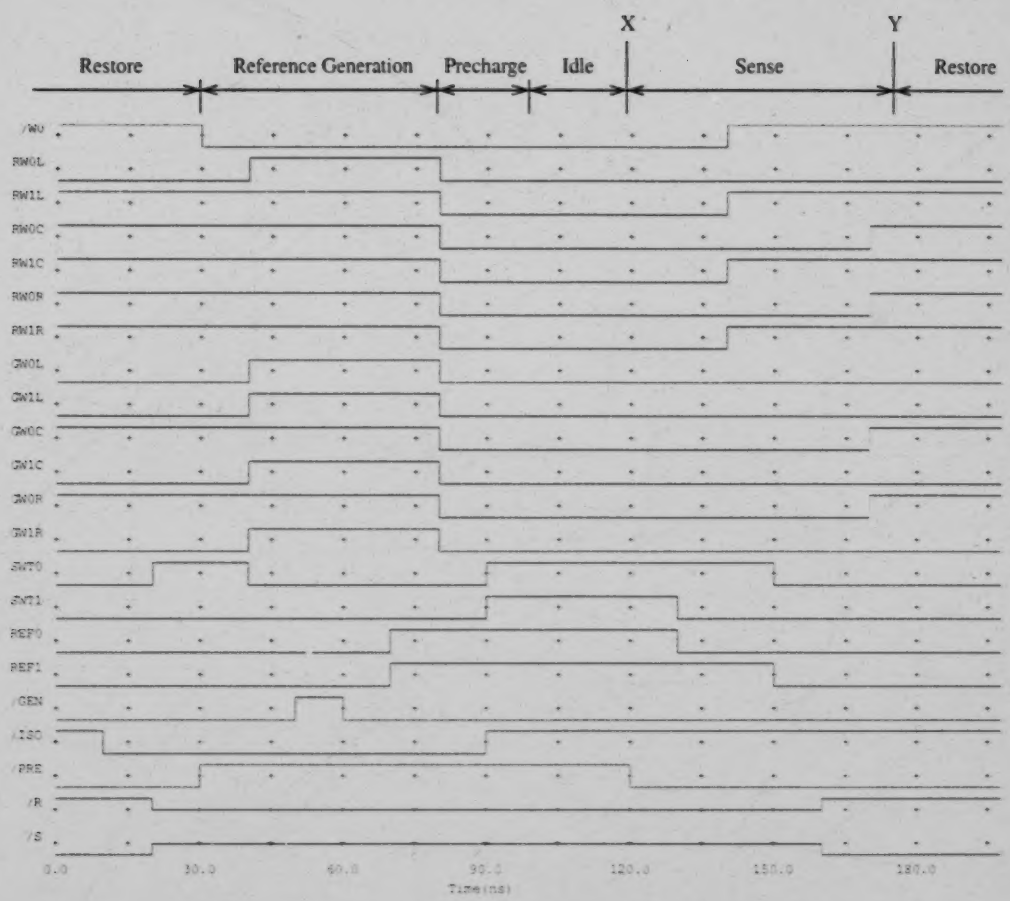


FIGURE 5.3. Control timing for the new MLDRAM scheme

used to completely balance out all sub-bitline capacitances and (b) there is no shorting operation between parallel sub-bitlines.

Referring to Figure 5.3, the restore occurs first by asserting SWT0 (t = 20 ns) to short the sub-bitlines together and then by de-activating the wordline (t = 30 ns) capturing the voltage in the memory cell. The restore is operation complete. Now the reference voltages must be generated and stored in the reference cells. This is done by asserting the signals GWxx and RWxx (i.e. all signals beginning with either GW or RW). The capacitance on all sub-bitlines is now matched and balanced. In fact, the careful use of the reference and generate wordlines ensures that capacitances that participate in charge sharing are

properly matched throughout. With all sub-bitlines isolated from each-other and all having the same capacitance, the GEN signal is pulsed (t = 50 ns to t = 60 ns). The sub-bitlines are charged to their respective voltage values through the connections to the VDC port on each sub-bitline shown in Figure 5.2. Following this the REF0 and REF1 signals are asserted (t = 70 ns), shorting the sub-bitlines together causing the exact reference level to be created in each of the L, C, and R zones. Shortly thereafter the reference voltages are captured in the reference cells by de-asserting all reference wordlines. Table 5-2 shows how this occurs for each group of sub-bitlines (L, C, and R). Through this novel method of charge sharing the reference voltages are exactly created.

At the same time that the reference wordlines are deactivated the generate wordlines are deactivated to prepare for bitline precharging. The circuit returns to the inactive precharge state (after t = 90 ns). As shown on the diagram, the SWT0 and SWT1 switches are turned on and the REF0 and REF1 switches are turned on. A quick application of conservation of charge reveals that the 18 sub-bitlines (9 sub-bitline pairs) all tend to $\frac{1}{2}V_{DD}$. This is the precharge voltage. This easy precharging is unlike [1] where there is a serious precharging problem. In [1] the worst case is where all cells are restored to the 01 or 10 state. In that case the precharge voltage source, which is usually a current limited on-chip source, is heavily loaded with the burden of charging every bitline back to V_{BLP}.

TABLE 5-2. Reference generation for the left, center and right sub-bitlines

Sub-Bitline	VDC Voltage	Final Voltage
TL	V_{DD}	$\frac{5}{6}V_{DD}$
ML	$\frac{1}{2}V_{DD}$	$\frac{5}{6}V_{DD}$
BL	V_{DD}	$\frac{5}{6}V_{DD}$
TC	$\frac{1}{2}V_{DD}$	$\frac{1}{2}V_{DD}$
MC	$\frac{1}{2}V_{DD}$	$\frac{1}{2}V_{DD}$

TABLE 5-2. Reference generation for the left, center and right sub-bitlines

Sub-Bitline	VDC Voltage	Final Voltage
BC	$\frac{1}{2}V_{DD}$	$\frac{1}{2}V_{DD}$
TR	V_{SS}	$\frac{1}{6}V_{DD}$
MR	$\frac{1}{2}V_{DD}$	$\frac{1}{6}V_{DD}$
BR	V_{SS}	$\frac{1}{6}V_{DD}$

From the idle precharge state a random access may occur. Assume again that the cells along wordline W0 are accessed. Because W0 is even, it associates with the true bitline and thus the SWT1 switch is de-asserted leaving the true bitlines connected horizontally {L to C to R} through SWT0 (t = 130 ns). At the same time, the REF0 switch is de-asserted leaving the complementary bitlines shorted vertically {T to M to B}. Shortly thereafter, the reference wordlines RW1L, RW1C and RW1R and the wordline W0 are asserted together (t = 140 ns). Each of the three memory cells (in T, M and B) are connected to one horizontal group of three sub-bitlines. As well, each of the three reference cells are connected to one vertical group of three complementary sub-bitlines. Because both the memory cell charge and the reference cell charge experience the same charge sharing circumstances and resulting signal attenuation, the sense amplifiers directly compare the charge stored in the memory cell to the charge stored in the reference cell.

The charge from the addressed cell is diluted across three sub-bitlines and one cell through two switches. The reference signals are each shared from one reference cell to three sub-bitlines and one cell through two switches. The exact capacitive symmetry effectively cancels out previously ignored but important higher-order effects. Each and every signal has equal charge injection affecting it. After the charge sharing is complete the remaining switches, REF1 and SWT0, are de-asserted leaving each sub-bitline totally isolated (t = 150 ns). The sense amplifiers are turned on and the two bits are sensed in parallel for each of the bitlines giving the scheme a low access time.

One final operation is performed just after sensing. That is, in order to prepare for restore, the reference and generate wordlines that do not share the same sub-bitline as the wordline are asserted. In this case, RW0C, RW0R, GW0C and GW0R are asserted (t = 170 ns). This ensures that the restore operation will consist of exactly one sub-bitline and one cell capacitance per sense amplifier thus conforming to the exact capacitance matching needed for proper restore

This MLDRAM scheme can be expanded to include any number of levels. The above description uses a three-by-three array of sub-bitlines for four levels, but it could use any square number of sub-bitlines. The number of levels, N, in an MLDRAM using this scheme requires an N-1 by N-1 array of sub-bitlines with the reference levels spaced evenly according to Eq. 1.5, repeated below for convenience.

$$V_{REF} \in \{1, 3, 5, \ldots, 2N-3\}\frac{V_{DD}}{2(N-1)} \tag{5.1}$$

The REF and GEN signals would connect the N-1 sub-bitlines in the vertical direction and the SWT0 and SWT1 signals would connect the N-1 sub-bitlines in the horizontal direction, as they do for the four-level case. The references would be generated in the same fashion and the connections to the VDC transistors in each vertical column (not to be confused with the DRAM column) would vary so that each of the N-1 reference cells in the horizontal direction would contain one of the references given by Eq. 5.1. The vertical N-1 VDC connections for any vertical column of sub-bitlines can be determined for any reference by the following.

Each VDC can be connected to either V_{DD}, V_{SS} or V_{BLP}. These voltages, when charged onto the sub-bitlines, must charge share to the reference voltage:

$$V_{REFi} = m_i\frac{V_{DD}}{2(N-1)}, \text{ where } m_i \in \{1, 3, 5, \ldots, 2N-3\} \tag{5.2}$$

(this is simply another form of Eq. 5.1)

The coefficient m_i can be created through charge sharing each of the sub-bitlines together according to the following equation:

$$m_i = \sum_{j=1}^{N-1} k_j \tag{5.3}$$

where k_j can be 0 or 1 or 2. After determining each k_j for each m_i to satisfy Eq. 5.2 and Eq. 5.3 simultaneously, the VDC connections can be made for each of the N-1 sub-bitlines by connecting one sub-bitline to V_{DD} for each k_j = 2, one sub-bitline to V_{BLP} for k_j = 1, and one sub-bitline to V_{SS} for each k_j = 0. For example, consider a 7-level MLDRAM (N=7). Then the six corresponding reference voltages can be determined by the entries in Table 5-3, which shows all of the k_j values for each of the reference voltages.

TABLE 5-3. Values of k_j for each reference voltage when N = 7.

i	m_i	k_1	k_2	k_3	k_4	k_5	k_6	V_{REFi}
1	1	0	0	0	0	0	1	$\frac{1}{12}V_{DD}$
2	3	0	0	0	1	1	1	$\frac{3}{12}V_{DD}$
3	5	0	1	1	1	1	1	$\frac{5}{12}V_{DD}$
4	7	2	1	1	1	1	1	$\frac{7}{12}V_{DD}$
5	9	2	2	2	1	1	1	$\frac{9}{12}V_{DD}$
6	11	2	2	2	2	2	1	$\frac{11}{12}V_{DD}$

5.2 Discussion

One advantage of the new MLDRAM design is that, to a first order approximation, it is not susceptible to signal errors due to unbalanced charge injection. Each signal transition

in the timing that causes charge to be injected unequally (for example, wordline activation) is balanced by another switch of equal charge injection (for example, reference wordline activation). Another advantage is speed. The access time should be small because the references are in place at the start of the cycle and both bits are sensed and available at the same time. Also, the circuit uses only local reference and restore voltage generation thus the reliability should be high.

The main disadvantage of this design is the area overhead required. For example, if a DRAM has 256 wordlines along one bitline then a comparable MLDRAM of this type would have 85 wordlines per sense amplifier. Thus for the storage capacity of 85 wordlines there is one sense amplifier, 4 extra rows, and a switch region for the REF, GEN, and SWT signals. The addition of this circuitry drastically reduces the density gain that MLDRAM offers. If the density gain is reduced too much by all of the required area overhead, it may not be more economical than DRAM.

CHAPTER 6: SIMULATION STUDY OF FOUR MLDRAM CIRCUITS

The several proposed MLDRAM schemes differ in the techniques used to store the four analog cell voltages and/or the techniques used to sense and restore the cell signals. It is difficult to compare these schemes directly because they were described for different processes at different times. To better gauge the real advantages and disadvantages of the alternative schemes, it is important to compare them fairly using the same process parameters, array size, and device models. This chapter presents a simulation study that attempts to provide a fair comparison of the MLDRAM schemes described by [7], [1], [8], and the new scheme proposed in Chapter 5. We did not consider a 4-level version of the 16-level MLDRAM scheme proposed by [6] because fundamental limitations (e.g. long access time) render it uncompetitive with the other designs.

6.1 Experimental Setup

6.1.1. Controlled Parameters

The simulations were performed using a process model based on a composite of typical 250 nm CMOS processes. For each scheme, a minimum of 256 wordlines and one bitline pair were modeled (three pairs were required in the new scheme). For a control case, a DRAM bitline pair was modeled and simulated. This can be used to compare the various advantages and disadvantages of MLDRAM in general with each other and with conventional DRAM.

Ideal capacitors were used for the cells and a distributed RC model was used to model the bitline parasitic resistance and capacitance. For all of the schemes, I used the same basic cell array model, shown in Figure 6.1. This fundamental unit of four cells requires that scheme [7] and the proposed scheme have greater than 256 wordlines since the number of sub-bitlines is not a multiple of 64. It should be noted that the C_{GS} of the cell access transistors and the switches account for most of the bitline capacitance. Extra capacitance (CP2 in Figure 6.1) is added as a way to control the cell-to-bitline capacitance

ratio and is intended to only loosely resemble the bitline poly capacitance encountered in a DRAM process.

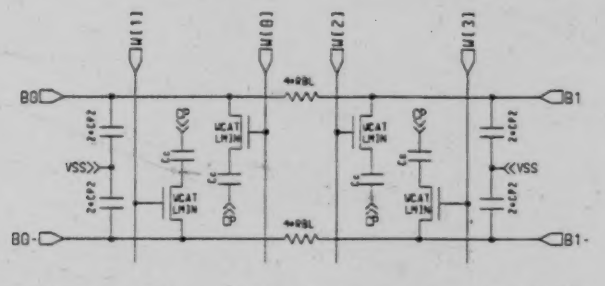


FIGURE 6.1. The model for four memory cells (4MC).

The same sense amplifier model was used for all simulations. Each scheme uses a generic DRAM sense amplifier as a building block so this is a parameter that can be controlled without biasing the performance of any scheme. The sense amplifier used in the simulations is shown in Figure 6.2. To get realistic sensing times in a DRAM environment, a resistor was added in series with each of the SAP and SAN- voltage sources to provide transient behavior similar to a worst case in a conventional DRAM. Then, this model for the SAP and SAN- voltage supplies was used for each simulation.

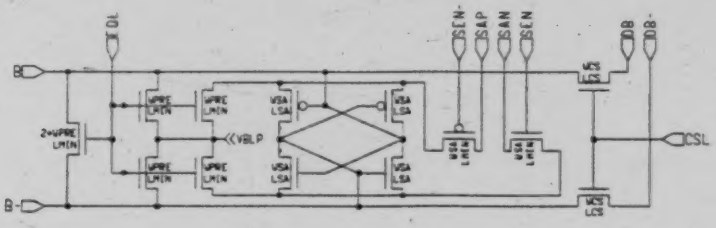


FIGURE 6.2. The model used for the sense amplifiers.

Controlled parameters, including resistances, capacitances and transistor sizes are given in Table 6-1.

TABLE 6-1. Simulation Conditions

Parameter	Description	Value
V_{DD}	Main supply voltage	2.50 V
V_{BLP}	Bitline precharge	1.25 V
V_{CP}	Cell capacitor common node	1.25 V
V_{BOOST}	High value for boosted NMOS switches	3.50 V
T	Temperature	27 C
t_r, t_f	Control signal rise and fall time	2.0 ns
t_w	Wordline rise and fall time	5.0 ns
t_{skew}	Time allowance between edges	0.5 ns
RBL	Bitline resistance per cell	25 ohms
C_{P2}	Bitline poly parasitic capacitance per cell	100 aF
Cc	Cell capacitance	25 fF
Cb/Cc	Full Bitline to cell capacitance ratio	~5.33
WCAT	Cell access transistor width	300 µm
LMIN	Transistor minimum length	250 µm
WSA	Sense amplifier transistor width	2.0 µm
LSA	Sense amplifier transistor length	0.5 µm
WSW	NMOS switch width	1.0 µm
WPRE	Precharge transistor width	0.5 µm

6.1.2. MLDRAM Schematics

The SPICE simulations were performed as described in their respective papers. Wherever possible, timing waveforms and circuit configurations are true to their paper descriptions, but some modifications were required to fit the schemes into the experimental setup. For example, the sense amplifier in [7] is slightly different from the one that we used (Figure 6.2), but both work for the MLDRAM scheme. This section describes the schematics used for each scheme in the study. Specific details and assumptions are noted, as well as discrepancies between the schematics and their respective paper descriptions.

For MLDRAM, as in regular DRAM, read and write operations occur once the sense amplifiers are turned on. The modeling and operation of the data-bus and column access was not included in the simulations. Each scheme could be equipped with the same data-bus interface and so this circuit would not be a differentiating factor among the MLDRAMs being studied.

6.1.2.1 Furuyama's MLDRAM

Figure 6.3 shows the schematic for Furuyama's scheme. The three equal-length sub-bitlines are shown, each with a different VDC parameter connected to the dummy (reference) cells through DCP-. The value of the reference voltages VDCA, VDCB and VDCC are not stated in the paper. The following circuit analysis was used to determine the reference voltages.

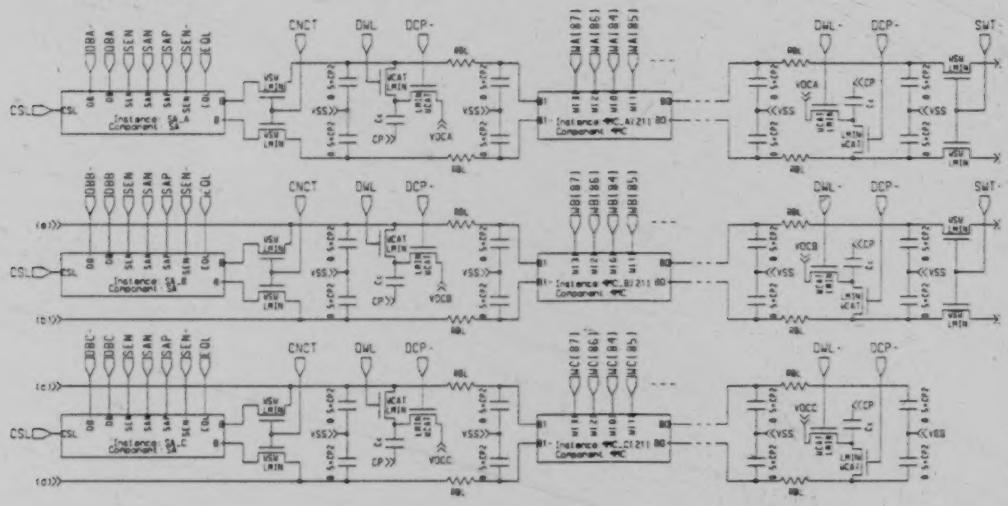


FIGURE 6.3. Schematic used for simulating Furuyama's MLDRAM [7].

When the wordline is activated and the cell voltage V_C is shared from the cell capacitor C_C onto the three sub-bitline capacitances C_B, the resulting signal on all three the true bitlines is

$$V_S = \frac{V_C C_C + V_{BLP}(3C_B)}{C_C + 3C_B} \tag{6.1}$$

where the cell voltage may be one of the allowed multilevel voltages:

$$V_C \in \{0, 2, 4, 6\}\frac{V_{DD}}{6} \tag{6.2}$$

The reference voltage V_R placed on the complementary sub-bitline can be expressed as an ideal reference cell voltage V_{REF} that experiences the same charge-sharing function:

$$V_R = \frac{V_{REF} C_C + V_{BLP}(3C_B)}{C_C + 3C_B} \tag{6.3}$$

where the reference cell voltage may be one of the ideal reference voltages:

$$V_{REF} \in \{1, 3, 5\}\frac{V_{DD}}{6} \tag{6.4}$$

It is not sufficient to set the actual reference voltage, V_{DC}, equal to one of the three V_{REF} values for each of VDCA, VDCB, and VDC because the charge sharing function that the reference cells experience is not equal to that of Eq. 6.3. The actual reference voltage V_{DC} is charge shared with only one sub-bitline. Thus,

$$V_R = \frac{V_{DC} C_C + V_{BLP} C_B}{C_C + C_B} \tag{6.5}$$

Solving Eq. 6.3 and Eq. 6.5 for V_{DC} gives

$$V_{DC} = \frac{V_{REF} + 2V_{BLP}(1-K)}{3-2K} \tag{6.6}$$

where K is the capacitor ratio:

$$K = \frac{C_C}{C_C + C_B} \tag{6.7}$$

K was found using Eq. 6.5 and sweeping V_{DC} from V_{DD} to 0V. The simulation showed that K varies as a hyperbolic function of V_{DC} with K approaching infinite proportions as V_{DC} approaches V_{BLP}. Obviously the capacitance ratio does not change so drastically due to the nonlinear effects of the PN-junction parasitic capacitances that contribute to both C_C and C_B. However, Eq. 6.5 neglects charge injection effects. Charge injection effects are present and are caused by the DCP- switch and the reference wordline. Thus the working V_{DC} value must compensate for these effects if the reference signal to reflect Eq. 6.3. An approximate K value of 0.339 was determined by the K asymptote. Based on this K value, V_{DC} values were calculated from Eq. 6.5. From these starting guesses a series of simulations was performed to search for working V_{DC} values that not only reflect the difference in charge-sharing between the memory cell and the reference cell, but compensate for the charge injection effects of DCP-. The final V_{DC} values used in the simulation are shown in Table 6-2.

TABLE 6-2. V_{DC} voltages used for simulating Furuyama's MLDRAM

V_{DC} Reference Voltage	Result from Eq. 6.6 (V)	Final Compensated Voltage (V)
VDCA	0.891	0.860
VDCB	1.250	1.162
VDCC	1.609	1.521

It should be noted that the V_{DC} voltages are functions of the charge injection due to DCP-, and functions of the capacitor ratio K. Also, The reference signal V_R can be expressed as $V_R = \frac{1}{K} V_{DC}$ so as K decreases, the reference signal voltage (V_{DC}) becomes more sensitive to V_{DC}.

6.1.2.2 Gillingham's MLDRAM

The schematic for Gillingham's scheme is shown in Figure 6.4. It consists of two sub-bitlines of equal length, each with its own sense amplifier. Four dummy wordlines are added to each 128-cell sub-bitline, as required. The cross-connect switch matrix, shown in center of Figure 6.4, contains all the required switches. All the switch transistors are the same size except for EL and ER, which are double sized because they act as precharge devices.

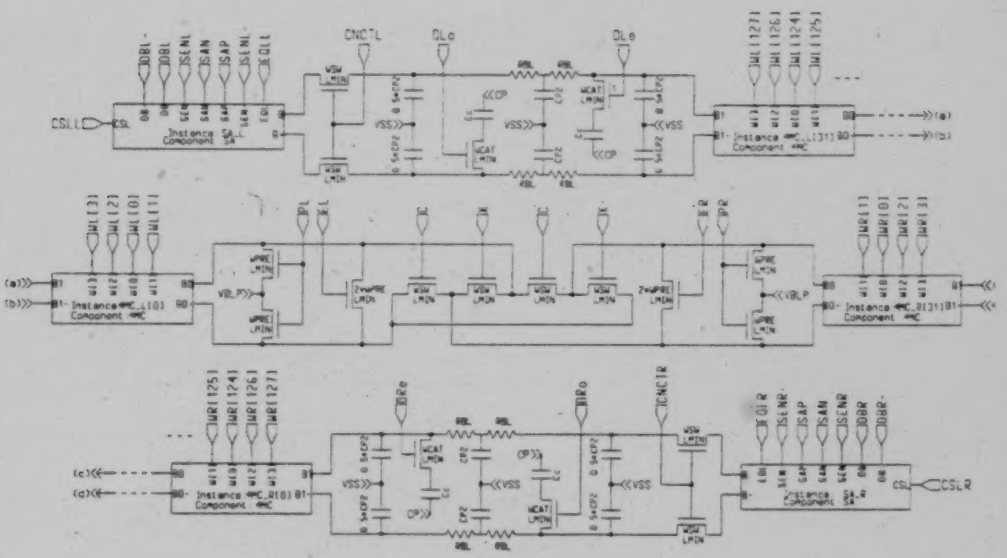


FIGURE 6.4. Schematic used for simulating Gillingham's MLDRAM [1].

6.1.2.3 Okuda's MLDRAM

Okuda's MLDRAM, shown in Figure 6.5, is adapted from the paper so that it contains the sense and restore scheme described in part II, sections A, B and C only [8]. Their implementation of this sense and restore scheme includes hierarchical bitlines to maximize density and offset-cancelling sub-sense amplifiers for density and immunity to transistor threshold mismatches. These circuits are external to the main MLDRAM scheme and would not be required for this hypothetical process (i.e. the sense amplifier fits in the pitch of a bitline pair). The main circuitry required for MLDRAM operation is the cross-connecting shunt capacitors, shown in the center of Figure 6.5. The shunt capacitors bisect the array into one sub-bitline of 84 wordlines and the other of 172 wordlines. This arrangement gives a sub-bitline capacitance ratio of approximately 2.05:1. This is the closest approximation of the required 2:1 sub-bitline capacitance achievable within the constraints of the simulation. This small discrepancy between the simulation model and the paper specification is negligible with respect to the parameters under study.

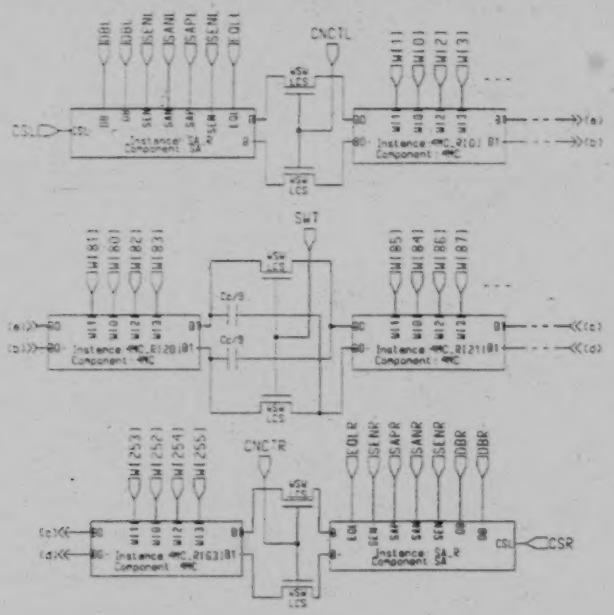


FIGURE 6.5. Schematic used for simulating Okuda's MLDRAM [8].

Okuda's paper states that shunt capacitors should have a capacitance of $\frac{C_C}{3}$, but they are shown in Figure 6.5 to have a capacitance of $\frac{C_C}{9}$. This different result comes from the following analysis. When the wordline is activated and the cell capacitor (C_C) charge is shared across both sub-bitlines, the differential signal presented to the sense amplifiers is

$$\Delta V_S = \frac{V_C C_C}{C_C + 3C_B}, \text{ where } C_B \text{ is the capacitance of the smaller sub-bitline.} \quad (6.8)$$

This differential is disturbed by ΔV_{REF} when the MSB is sensed because of the two shunt capacitors (C_X). Thus the total signal, ΔV is

$$\Delta V = \Delta V_S + \Delta V_{REF} \quad (6.9)$$

The ideal value of ΔV_{REF} is

$$\Delta V_{REF} = \left(\frac{2}{3}\right)\frac{C_C(V_{SIGN} - V_{BLP})}{C_C + 3C_B} \quad (6.10)$$

where V_{SIGN} is V_{DD} for a sign bit of 1, and V_{SS} for a sign bit of 0. The actual value of V_{REF} is calculated in terms of C_X:

$$\Delta V_{REF} = \frac{2C_X(V_{SIGN} - V_{BLP})}{C_B} \quad (6.11)$$

This assumes that $C_B \gg C_C$ because the activated wordline may be in the region of the smaller sub-bitline and so C_C would appear in the denominator of Eq. 6.11. It also assumes that $C_B \gg C_X$.

Eq. 6.10 and Eq. 6.11 can be solved for C_X:

$$C_X = \frac{1}{9}C_C \tag{6.12}$$

This new result of C_X allowed for correct MLDRAM operation. The assumptions used to create this result cause small signal errors that could be corrected slightly with a more through second-order analysis. However, it is not possible to arrive at one ideal C_X value because of the two distinct sensing cases: if the wordline is less that 84 then C_C plays a role in charge sharing, but if the wordline is greater or equal to 84 then C_C does not participate. Thus, before a more detailed approach is taken, one may consider the use of dummy wordlines to remedy this dilemma. In any case, the small signal error can be neglected for the purposes of this simulation study.

6.1.2.4 The New MLDRAM

The final MLDRAM scheme in this study is the new scheme proposed in Chapter 5. The schematics for this scheme are shown in Figure 6.6 and Figure 6.7. Although all of the other schemes have only a single bitline-pair replicated to form the array, this scheme requires three bitline-pairs for proper operation. The circuit was constructed as required by the description in Chapter 5. As in Furuyama's scheme, the sub-bitlines are split into three equal parts which contain at least 256 wordlines in total. The extra wordlines (there are 264) are considered to be overhead necessary for the scheme. Each of the three bitline pairs is powered by different SAP and SAN- voltage sources, each with its own source impedance equal to the impedance used in all of the other schemes. This allows for fair speed and power comparisons between individual bitlines in all schemes.

6.1.3. Simulation Methodology

Simulations were performed in two steps. First, each scheme was simulated with relaxed timing to allow nodes to reach steady state between switching operations. After the steady state values were determined, the timing was optimized by compressing the operations in time as much as possible without compromising MLDRAM operation or

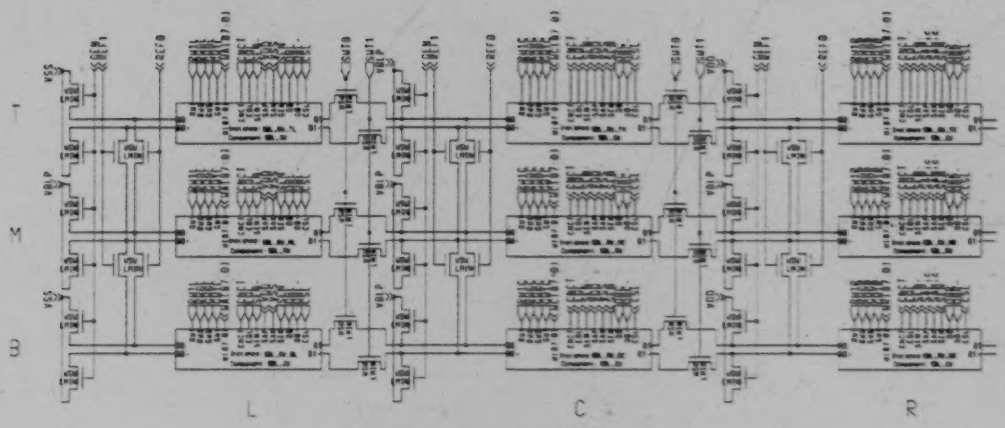
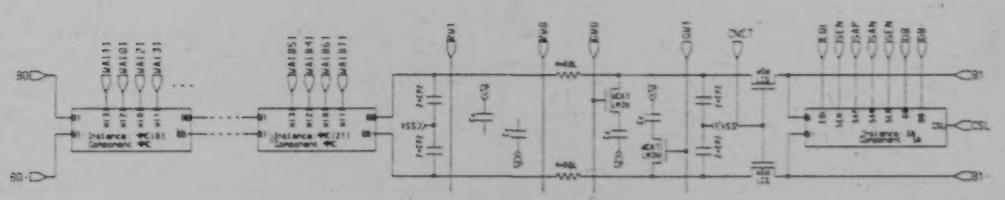
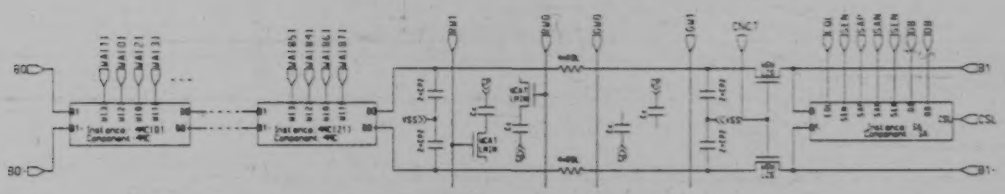


FIGURE 6.6. Schematic used for simulating the proposed MLDRAM (Chapter 5).



Reference sub-bitline (SBL-RW)



Generate sub-bitline (SBL-GW)

FIGURE 6.7. Sub-bitline Schematics for the proposed MLDRAM (Chapter 5)

signal quality. The criteria used to optimize the timing are shown in Table 6-3. Wherever a charge sharing operation was performed, the worst case of the nodes involved was allowed to meet the criterion before the next event in the critical timing path. On top of this

convergence time allowance, a constant delay, *tskew* (Table 6-1), was added between each edge to accommodate any skew between signal edges that might occur in a real chip.

TABLE 6-3. Timing Optimization Criteria

Switching operation	Criteria
Switch turning off	None (only rise/fall time + t_{skew})
Charge sharing	99% of their steady state value
Sensing	Within 5mV of the power supply voltage
Precharging	Within 1mV of V_{BLP}

Once the optimal timing for each circuit was found, each was simulated for access time, cycle time and energy.

6.2 Results

Our results are summarized in Table 6-4. Sense amplifier (SA) activation is the time when the SEN signal (Figure 6.2) is turned on. The bitline (BL) 0.5 V split is the time it takes for the sense amplifier to develop a differential voltage of 0.5 V on the bitlines (or sub-bitlines). This parameter is important because the different sensing operations take different times and the 0.5 V split gives a fair indication of the delay required for a read operation. For Gillingham's and Okuda's schemes the MSB and LSB are available at different times so there are separate rows in Table 6-4 for the two bits. The cycle time is measured from the first event to the time when the final precharge criterion (from Table 6-3) is met. The energy per bit was measured by integrating the instantaneous power over the cycle time. The details of the energy calculations follow.

For V_{DD}, the energy was calculated using Eq. 6.13. The same formula was used for the sense amplifier power supply energy.

$$E_{VDD} = \int_t V_{DD} I_{VDD} \tag{6.13}$$

For the V_{BLP} power supply, the energy was calculated using the formula

$$E_{VBLP} = \int_t V_{DD} I_{VBLP} \tag{6.14}$$

because in commodity DRAMs V_{BLP} is generated on-chip so the current drawn from V_{BLP} is supplied through V_{DD}. Thus, Eq. 6.14 includes that portion of the energy consumed by the V_{BLP} regulator on-chip as well as that consumed within the sense and restore circuitry.

For each of the control switches that swing between V_{DD} and V_{SS}, the formula used was

$$E_{CONTROL} = \int_t V_{DD} max(I_{CONTROL}, 0) \tag{6.15}$$

Where the function *max(x,y)* is defined as

$$max(x, y) = x \text{ for } x > y \text{ and } max(x, y) = y \text{ for } x < y \tag{6.16}$$

Where boosted signals were used, the voltage element in the power equation was $2V_{DD}$. The boosted voltages are not normally available at the chip pins so signal boosting is done via a charge pump. The current drawn from a charge pump originates from the V_{DD} supply. Regardless of the boost voltage, the charge pump consumes power equal to $V_{DD}I$ so the total power used for boosted signals is $2V_{DD}I$. Thus, the formula used to calculate boosted signal energy was

$$E_{BOOST} = 2\int_t V_{DD} max(I_{BOOST}, 0) \tag{6.17}$$

The final row in Table 6-4 is representative of the control complexity required for the circuits. The number of edges in the critical timing path is roughly the number of sequential timing elements required on-chip.

For each of the schemes, all of the energy contributions were summed and the final result was divided by the number of bits sensed and restored.

TABLE 6-4. Simulation Results

Attribute	DRAM	Furuyama	Gillingham	Okuda	New
SA activation, 1st bit (ns)	8.00	14.06	18.33	15.27	14.25
BL, 0.5V split, 1st bit (ns)	9.08	16.73	20.16	17.00	17.03
SA activation, 2nd bit (ns)	N/A	14.06	70.25	34.04	14.25
BL, 0.5V Split, 2nd bit (ns)	N/A	16.73	72.50	35.40	17.03
Cycle time (ns)	37.87	57.75	152.48	82.09	77.74
Energy per bit (fJ/bit)	809	1101	1490	767	1605
Number of edges in the critical timing path	6	9	22	11	15

An example of the sense and restore waveforms from the simulations is shown in Figure 6.8. All bitlines with all possible data values are plotted for each scheme, including the one-bit DRAM case. Each plot shows the full sense and restore scheme. Unlike the others, the new scheme is shown starting with the restore, then precharging, and finishing with the sensing. This was done to avoid assuming perfect reference cell signals at the start. Rather, the reference cell signals are generated and captured in the reference cells. The precharge follows just before the actual start of the sense and restore cycle at t=45 ns.

From examining the waveforms one can see that for Furuyama's and the new scheme there is only one sensing operation, indicative of the flash-style sensing. For Gillingham's and Okuda's schemes the two sequential sensing operations can be seen.

Another observation is that the quality and placement of the restore values varies somewhat. This is an indication of the degree to which the sub-bitlines used in the restore are matched and balanced, and the degree to which charge injection effects are cancelled.

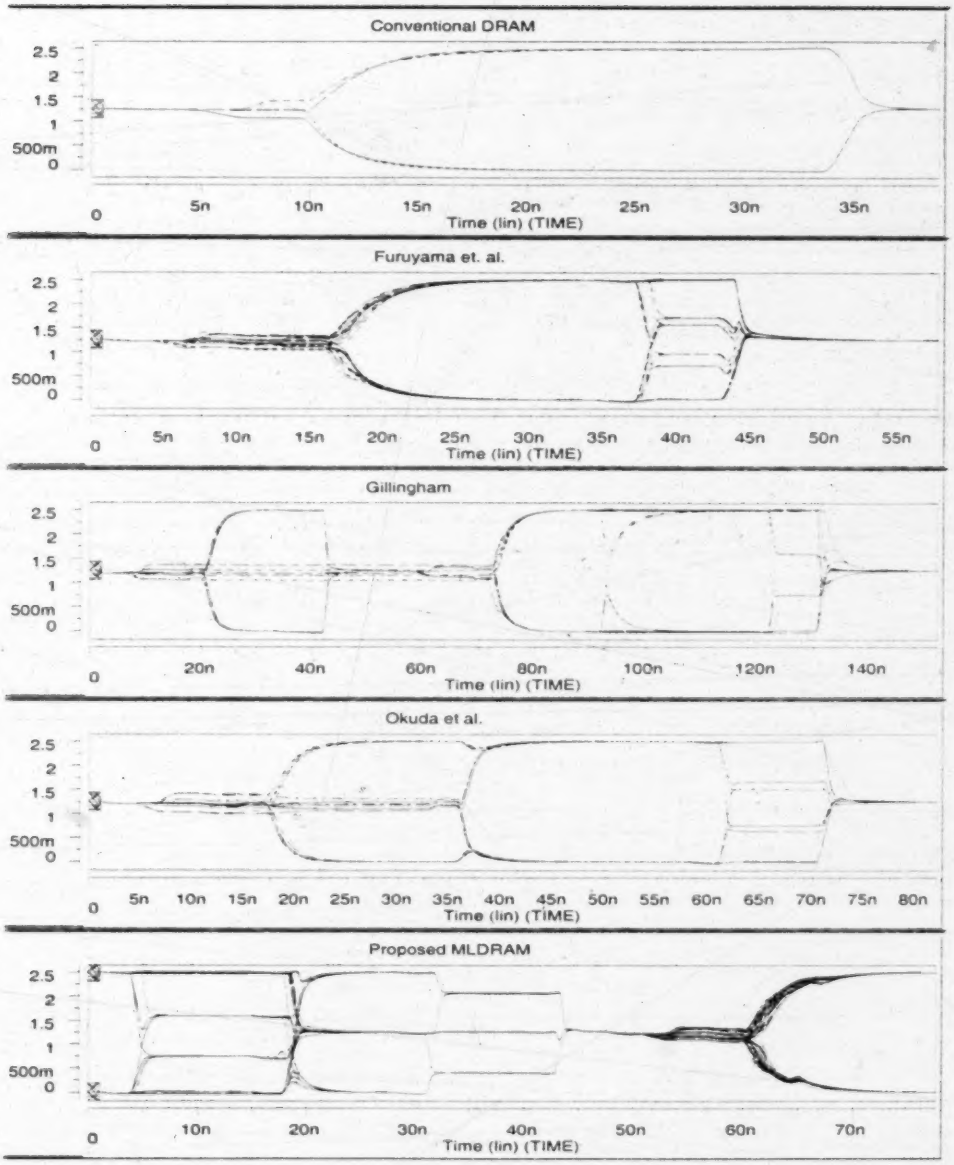


FIGURE 6.8. Sense and restore waveforms from the simulations

6.3 Discussion

The quantitative and qualitative aspects of the MLDRAM schemes are compared briefly and summarized below in Table 6-5, 1=Best Case and 5=Worst Case.

TABLE 6-5. MLDRAM Ranking

Attribute	DRAM	Furuyama	Gillingham	Okuda	New
Access time	1	2	5	4	3
Cycle time	1	2	5	4	3
Cycle energy per bit	2	3	4	1	5
Overall robustness	1	4	3	5	2
Area per bit	5	3	2	1	4
Control logic simplicity	1	3	5	2	4

As compared with conventional two-level DRAM, all MLDRAMs suffer in terms of speed. From the results it is clear that the parallel sensing used both in Furuyama's and the proposed scheme provide the fastest access and cycle times.

With the exception of Okuda's MLDRAM, the sense and restore energies per bit are all higher for MLDRAM than for DRAM. The reason for this is as follows. For Furuyama's and the proposed method there are three sense amplifiers that operate per cycle. For each sense amplifier, significant crowbar current leaks from VDD to VSS during the first few moments of sensing, thus increasing the energy used despite the fact that approximately the same capacitance is being charged to the same voltage as in a DRAM. This is also true for the other MLDRAM schemes, but they have fewer sense amplifiers leaking charge to ground. The other reason for increased energy per bit is the use of control switches. The energy used to switch devices on and off was significant, especially for those signals which needed to be boosted above VDD.

The sense and restore energy of Gillingham's scheme is higher than Furuyama's because there are more charging operations per cycle. Specifically, during restore in Gillingham's scheme one sub-bitline is completely charged from VSS to VDD. As well, the control switching is the most complex so the energy required for this is highest. The

energy of the proposed method is higher than Furuyama's because the reference generation procedure requires extra energy to charge the required bitlines to VDD. The energy of Okuda's scheme is actually lower than that of DRAM because it has minimal extra control circuitry and sparse use of boosted control signals in order to recover two bits.

In terms of area, Okuda's scheme has the least increase over DRAM because it uses a minimal amount of circuitry and only two sense amplifiers. Gillingham's scheme is a close second to Furuyama's scheme only because of the extra area used by the switch matrix. Furuyama's and the proposed scheme have the worst overall area gain because they use three sense amplifiers per bitline.

In terms of robustness, the new scheme is ranked the best among the MLDRAMs because of the use of local reference generation and local charge sharing based restore. In terms of signal quality it is better than Gillingham's scheme because it has balanced control switch usage, which means charge injection errors are not a concern. Furuyama's has good robustness because it uses charge sharing for the restore voltage, but it may suffer from the accuracy and impedance of the voltage supplies that are required to charge the reference cells. Also, the reference voltage supplies in Furuyama's scheme are sensitive to bitline capacitances and charge injection effects, both of which are undesirable for a robust design. Okuda's scheme ranks last because, although it uses charge sharing, LSB sensing depends heavily on the capacitor used to generate the reference. The exact size of this capacitor is critical so this raises reliability concerns.

The final criterion on which the MLDRAMs are compared is the control complexity. In general, the more complex the control timing, the more room there is for skew errors that affect the soft error rate. Furthermore, the more control logic that is required, the more area and power is consumed when generating the signals and driving the long control wires. The ranking shown in Table 6-5 is given according to the number of edges required for the cycle.

6.4 Conclusions

MLDRAM has been proposed by several authors as a way of increasing storage density without reducing the physical feature size. As a result of the disadvantages of reduced noise margins, increased circuit complexity and longer access and cycle times, MLDRAM has yet to enter production parts. The simulation study clarifies the real strengths and weaknesses of proposed MLDRAM schemes as well as the new MLDRAM from the previous chapter.

CHAPTER 7: CONCLUSIONS

There are many challenges and obstacles that must be faced before MLDRAM is used in an industrial design. Each of the past MLDRAM schemes considered in this thesis has approached the multilevel sense and restore problem in a novel and creative way, and each has added to the growing pool of potential MLDRAM solutions. It is my hope that the research presented in this thesis will aid future MLDRAM researchers and designers by giving insight into the many practical problems that must be overcome. The main contributions of this thesis are the implementation and test of Gillingham's MLDRAM, the introduction of a new MLDRAM scheme, and a comparative simulation study of the new MLDRAM scheme alongside past schemes. Also, the appendix material provides a method of charge injection analysis and simulation that casts light on the sources of signal errors in MLDRAM sensing.

7.1 MLDRAM Chip Designs

The MLDRAM test chip, based on Gillingham's design, was fabricated using TSMC's 0.35 µm CMOS ASIC process. The design was based on MOSAID's HDRAM, an embedded ASIC DRAM. The HDRAM design was modified to include 8 bitline pairs and 128 wordlines for a total of 1024 memory cells. Also, extra sense amplifiers and cross-connect circuitry were added to the core as required for MLDRAM operation. A main feature of the test chip is the availability of most of the control signals at the chip pads. The onus is then on the tester to supply all core control timing edges. Thus the tester has the flexibility to operate the chip using various timing control sequences. For example, the chip was tested first as a conventional 1-bit/cell DRAM and later as a 2-bit/cell MLDRAM. Two versions of the MLDRAM test chip were implemented. The first had a serious place-and-route error which rendered MLDRAM operation impossible. However, the test chip still contained the required functionality for the right cell array to be operated and evaluated as a 1-bit/cell DRAM. The second version of the test chip did not contain this error.

7.2 MLDRAM Test Chip Characterization

Testing of the MLDRAM test chips was carried out using an IMS LogicMaster XL tester. Although the tester was not intended for memory testing, it still provided the necessary functionality. Test vectors were generated conventionally using a flexible C++ program. The program included de-scrambling functions so that proper march tests could be performed.

The first test chip was tested in 1-bit/cell DRAM mode. The chip functioned properly and passed a standard 5n march test. By using the tester's programmable delays, I was able to fine-tune all of the DRAM control timing edges to reach an optimal access time. Although this chip did not offer any proof of Gillingham's MLDRAM scheme, testing it was valuable as it demonstrated the DRAM functionality of the design, leaving only the MLDRAM aspects to be tested.

The second test chip was first tested in 1-bit/cell DRAM mode. After identifying a databus precharging problem, an alternate timing scheme was applied to correct the problem. Then, DRAM functionality was verified. MLDRAM testing began immediately thereafter. During MLDAM testing, some pin labelling problems were discovered and corrected. Proper MLDRAM functionality was demonstrated although some errors in LSB sensing (when data = 10) remain for some of the MLDRAM columns. Nevertheless, the functionality of Gillingham's scheme was verified in silicon.

7.3 An Improved MLDRAM Scheme

A new scheme was proposed that offers very high robustness. The scheme overcomes the long access time of Gillingham's design and improves upon its robustness by using only exactly matched and balanced charge-sharing operations for both the restore operation and reference generation. The speed of the new scheme is a result of the parallel sensing technique that was used by Furuyama *et al.* The area overhead of the new MLDRAM is slightly more than that of Furuyama's design but the scheme still offers

sufficient density gain over DRAM. A U.S patent has been filed to cover the new MLDRAM scheme.

7.4 MLDRAM Simulation Study

In order to fairly compare past designs with each other and the proposed scheme, a simulation study was designed and conducted. Each design was simulated under equal conditions to compare access time, cycle time and cycle energy. Also, MLDRAM robustness, area overhead and timing complexity were determined and compared. With respect to DRAM, all of the MLDRAM schemes required more time for access, had a longer cycle time and required more energy per bit. However, each MLDRAM scheme has a higher bit density than DRAM, as expected. Furuyama's MLDRAM had the best MLDRAM access time, followed closely by the proposed new scheme. However, Furuyama's scheme is not robust and it requires more area overhead than the other schemes. Gillingham's scheme was found to be robust and it boasts less area overhead that Furuyama's scheme, but it is the slowest and most complex of the MLDRAM schemes. Okuda's scheme offers the best density (least area overhead), but it is also the least robust due to its sensitivity to inevitable process variations. The proposed scheme is comparable in speed to Furuyama's and it is very robust, but it has the worst area overhead penalty of all the MLDRAM schemes.

7.5 Future Work

Remaining MLDRAM research tasks are as follows. First, a silicon implementation of the proposed MLDRAM scheme is underway at the university of Alberta. A new fault model and accompanying test needs to be designed for testing the chip. Also a detailed sensitivity analysis of the different schemes to realistic process parameter variations and operating conditions would quantify robustness and add numerical strength to the robustness arguments put forth in this thesis. More work should be performed on the economics of MLDRAM. Such work would need to quantify the extra overhead required by MLDRAM. An analysis of the cost of an MLDRAM part versus a DRAM part would

identify the economic advantages (or disadvantages) of using MLDRAM in both embedded DRAMs and commodity DRAMs.

BIBLIOGRAPHY

[1] P. Gillingham, "A Sense and Restore Technique for Multilevel DRAM," *IEEE Trans. on Circuits and Systems-II: Analog and Digital Signal Processing*, Vol. 43, No. 7, pp. 483-486, July 1996.

[2] M. Redeker, B.F. Cockburn, and D.G. Elliott, "Fault Models and Test Strategies for a Two-bit per Cell DRAM", *1998 IEEE International Workshop on Memory Technology, Design and Testing*, Aug. 23-25, San Jose, CA, U.S.A, pp. 7-12.

[3] D. A. Johns and K. Martin, "Analog Integrated Circuit Design," John Wiley & Sons, 1997.

[4] B. Prince, "Semiconductor Memories: A handbook of Design Manufacturing & Applications," 2nd Ed., John Wiley & Sons, 1996.

[5] K. Itoh *et al.*, "Limitations and Challenges of Multigigabit DRAM Chip Design," *IEEE J. Solid-State Circuits*, Vol. 32, No. 5, pp. 624-634, May 1997.

[6] M. Aoki *et al.*, "A 16-Level/Cell Dynamic Memory," *IEEE J. Solid-State Circuits*, Vol. SC-22, No. 2, pp. 297-299, April 1987.

[7] T. Furuyama *et al.*, "An Experimental 2-bit/Cell Storage DRAM for Macrocell or Memory-on-Logic Application," *IEEE J. Solid-State Circuits*, Vol. 24, No. 2, pp. 388-393, April 1989.

[8] T. Okuda *et al.*, "A Four-Level Storage 4-Gb DRAM," *IEEE J. Solid-State Circuits*, Vol. 32, No. 11, pp. 1743-1747, Nov. 1997.

[9] T. Kawahara *et al.*, "A High-Speed, Small-Area, Threshold-Voltage-Mismatch Compensation Sense Amplifier for Gigabit-Scale DRAM Arrays," *IEEE J. Solid-State Circuits*, Vol. 28, No. 7, pp. 816-823, July 1993.

[10] R. C. Foss, "Implementing Application Specific Memory," *IEEE International Solid-State Circuits Conference*, pp. 260-261, 1996.

[11] P. Gillingham *et. al.*, "A 768K Embedded DRAM for 1.244Gb.s ATM Switch in a 0.8 μm Logic Process," *IEEE International Solid-State Circuits Conference*, pp. 262-263, 1996.

[12] S.S Iyer and H.L. Kalter, "Embedded DRAM: Opportunities and Challenges," *IEEE Spectrum*, Vol. 36, No. 4, pp. 56-64, April 1999.

APPENDIX A: MEASURING SENSING THRESHOLDS USING HSPICE

A.1 A Perl Script for Invoking HSPICE

```
#! /usr/local/bin/perl
##
## automatically runs hspice simulation using runsp
## and outputs a set of plots of vt v.s. length
##
## usage: vtplot min_length max_length length_step
##              norminal_voltage voltage_step output_prefix
##               bin_step [probe] [probetime]
##
##    min_length     = minimum length in um
##    max_length     = maximum length in um
##    length_step    = length resolution in um
##    norminal_voltage = reference for evaluaing output logic
##    voltage_step   = initial voltage resolution in v
##    output_prefix = prefix of the set of output files
##    bin_step       = number of steps in binary search
##     probe           = selects one of the four probes for
determining vt
##    probetime      = the time for probing in ns (default =
230.00000)
##
##  Script Function:   THis script takes the array_asim
stimulus file and manipulates the
##  length of one transistor and the initial cell voltage.
It is designed to do a
##sweep of both variables to extract for each length the
value of the inherent
##   (systematic) offset in the sense restore scheme of
mldram.  The script starts
## with a guess as to where the first reference point should
be and runs hspice.
##It then steps it's guess up by "voltage_step" and re-runs
hspice.  If the two
##simulations produce the same result then a linear search
begins to find the curve.
##If both are above the curve, a step is taken down by
"voltage_step".  Likewise if
##b oth are below a step is taken up.  The relitive position
of the curve is
```

```
##determined by comparing the result of hspice to one of the
three possible
##output values (i.e. mldram1 0, 1.1, 2.2, 3.3 Volts).
##
##Once two points are found so that the curve lies between
them a binary search is
##applied.  The extent of the search is given by "bin_step."
The final resolution
##of the output is equal to the "voltage_step" divided by 2
to the power of "bin_step"
##Some playing around can minimize the number of times
Hspice is envoked but it
##it will be at least 2 to the power of "binstep" plus 2*the
number of lengths
##(as calculated by 1+(max_length-min_length)/length_step
).  The algorithm
##increases the length by length_step and re-runs starting
with a guess based on
## the outcome of the completed binary search
##
## note : a stim template file is required for this script to
work.
##          the template file is simply the stim with some
parameter files
##        setting to generic variables:
##
##            .param ln1 = %LENGTH%U
##            .param initial_voltage = $INIT_VOLTAGE$
##
## by Mandr, Dec 15, 97

@commandline = @ARGV;

## read from parameter list
$minlen = shift (@ARGV);
$maxlen = shift (@ARGV);
$lenstep = shift (@ARGV);
$v_threshold = shift (@ARGV);
$vistep = shift (@ARGV);
$prefix = shift (@ARGV);
$binstep = shift (@ARGV);
$probe = shift (@ARGV);
$probetime = shift (@ARGV);

$minlen ne ""  || die "Minimum length required\n";
$maxlen ne ""  || die "Maximum length required\n";
```

```
$minlen < $maxlen || die "Minimum length < Maximum
length!!\n";
$lenstep ne "" || die "Length step required\n";
$v_threshold ne "" || die "norminal voltage required\n";
$vistep ne "" || die "voltage step required\n";
$prefix ne "" || die "Output prefix not set\n";
$probe = 1 unless ($probe ne "");
$probetime = 230 unless ($probetime ne "");
$n_spice = 0;

print "\n";
print "running simulations using the following
parameters:\n";
print "length = (min ",$minlen,"U, max ",$maxlen,"U, step
",$lenstep,"U) \n";
print "norminal voltage = ",$v_threshold,"V\n";
print "inital voltage step = ",$vistep,"V\n";
print "output file is ",$prefix,"\.vtplot\n";
print "selected probe = ",$probe,"\n";
print "binary depth = ",$binstep,"\n";
print "\n";

open (curve, ">".$prefix.".vtplot");
print curve "# vtplot ";
foreach $argu (@commandline)
{
    print curve $argu, " ";
}
print curve "\n";
$power = 1; for ($i=0; $i<=$binstep; $i++) { $power *= 2; }
print curve "# voltage error = +- ", $vistep/$power, "\n";
close (curve);


opendir (curdir, ".");
while ($_ = readdir (curdir))
{
    $stimtemplate = $_ if (/\.stim\.template$/);
    $spfile       = $_ if (/\.sp$/);
    system ("rm -f $_") if (/^$prefix.*\.data$/);
}
closedir (curdir);

$stimtemplate ne "" || die ("cannot find a stim
template\n");
```

```
sub setrun
{
    local ($len, $vi) = @_;          # the parameters

    open (template_file, $stimtemplate);
    open (stim_file,
">".                  substr($stimtemplate,                    0,
index($stimtemplate,"\.template")));
    while (<template_file>)
    {
s/%LENGTH%/$len/g;
s/%INIT_VOLTAGE%/$vi/g;
print stim_file $_;
    }
    close (template_file);
    close (stim_file);


    ## FIXME : this part might need to be fixed if changing
probe time ##
    $grepstring = sprintf ("%10.5fn",$probetime);

    system ("rm -f $spfile");

    $n_spice+=1;
          print   "Invoking   Hspice   (",$n_spice,")   for
length=",$len,", Vi=",$vi,"\n";

    open (spiceoutput, "runsp |");
    @vout = ($grepstring);
    while (<spiceoutput>)
    {
if (/$grepstring/)
{
          chop;
    s/  */ /g;
    @_ = split;
          shift (@_);
          push (@vout, @_);
}
    }
    close (spiceoutput);

    # print @vout, "\n";
```

```
    $vf = @vout[$probe];
    $vf;
}

sub digitize
{
    local ($analog_v) = @_;

    ## convert millivolt back to volt
    $_ = $analog_v;
    if (/m/)
    {
$analog_v /= 1000;
    }

    $digital_v = 0;
    $digital_v = 1 if ($analog_v > $v_threshold);

        # print "analog = ",$analog_v,"V; digital =
",$digital_v,"V\n";

    $digital_v;
}



  # the initial high and low are first calculated
  $vthigh = $v_threshold + $vistep/2;
  $vtlow  = $v_threshold - $vistep/2;


  # loop until reached maximum length
  for ($len = $minlen; $len <= $maxlen; $len += $lenstep)
  {
    # at each length, search for the pair of vi before and
after the vt
    print "Taking upper guess at ",$vthigh,"V \n";
    $vfhigh = digitize (setrun ($len, $vthigh));
    print "Taking lower guess at ",$vtlow,"V \n";
    $vflow  = digitize (setrun ($len, $vtlow));

    while ($vfhigh == 0)
    {
        ## real curve is above vfhigh, search for new vthigh
        ## increment $vtlow as a side effect
```

```
        $vthigh += $vistep;
        $vtlow += $vistep;
        print "Taking a step up to ",$vthigh,"V \n";
        $vfhigh = digitize (setrun ($len, $vthigh));
    }

    while ($vflow == 1)
    {
        ## real curve is below vflow, search for new vtlow

$vtlow -= $vistep;
        $vthigh -= $vistep;
        print "Taking a step down to ",$vtlow,"V \n";
        $vflow = digitize (setrun ($len, $vtlow));
    }


    ## binary search to enhance resolution

    for ($i = 0; $i <$binstep; $i++)
    {
$vtmid = ($vthigh + $vtlow)/2;
print "checking midpoint ",$vtmid,"V \n";
$vfmid = digitize (setrun ($len, $vtmid));

## if mid point about curve then adjust vthigh else vtlow
($vfmid == 1) ?
    $vthigh = $vtmid :
    $vtlow = $vtmid ;
    }


    open (curve, ">>".$prefix.".vtplot");
    $outline = sprintf ("%f %f\n",
 $len, ($vthigh+$vtlow)/2);
    print curve $outline;
    close (curve);

    ## set initial guesses for next length to the average of
this one
    ## plus/minus half the step

    $vthigh= ($vthigh+$vtlow+$vistep)/2;
    $vtlow = $vthigh - $vistep;

  }
```

A.2 The HSPICE Netlist Files Used

A.2.1. The MLDRAM circuit Netlist

```
$ DANET output file </user/mldram1/sch_asim/array_asim/
hspice/array_asim.net> $
$ Written on 10-Jul-1997 at 16:10:42 $

.global vdd vcc vss ground vbb vpp vblp vcp

.SUBCKT comp_0
+ IN OUT
+ M=1
RXI_1995#1 OUT IN 156  m=1
CXI_2064#1 OUT GROUND 139.093166200076F m=1
CXI_1994#1 IN GROUND 139.093166200076F m=1
CXI_2062#1 IN GROUND 137.198340368702F m=1
CXI_2063#1 OUT GROUND 137.198340368702F m=1
* Cell /user/mldram1/sch_lib/custom/wire
.ENDS $ comp_0 $

.SUBCKT comp_3
+ D G S
+ M=1
MXMO D G S VSS mn W=2.1U L=0.35U AD=1.785P AS=0P PD=3.8U
PS=0U NRD=0.238095238095238 NRS=0.238095238095238 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nmos
.ENDS $ comp_3 $

.SUBCKT comp_4
+ D G S
+ M=1
MXMO D G S VSS mn W=0.5U L=0.35U AD=0.425P AS=0.425P PD=2.2U
PS=2.2U NRD=1 NRS=1 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nxfer
.ENDS $ comp_4 $

.SUBCKT comp_5
+ B D G S
+ M=1
MXMO D G S B mp W=1U L=0.35U AD=0.85P AS=0.85P PD=2.7U
PS=2.7U NRD=0.5 NRS=0.5 M=1
* Cell /user/mldram1/sch_lib/comp_analog/pxferb
.ENDS $ comp_5 $

.SUBCKT comp_6
```

```
+ B D G S
+ M=1
MXMO D G S B mp W=2.8U L=0.7U AD=2.38P AS=0P PD=4.5U PS=0U
NRD=0.178571428571429 NRS=0.178571428571429 M=1
* Cell /user/mldram1/sch_lib/comp_analog/pmosb
.ENDS $ comp_6 $

.SUBCKT comp_7
+ D G S
+ M=1
MXMO D G S VSS mn W=0.7U L=0.35U AD=0.595P AS=0P PD=2.4U
PS=0U NRD=0.714285714285714 NRS=0.714285714285714 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nmos
.ENDS $ comp_7 $

.SUBCKT comp_8
+ D G S
+ M=1
MXMO D G S VSS mn W=1.4U L=0.35U AD=1.19P AS=1.19P PD=3.1U
PS=3.1U NRD=0.357142857142857 NRS=0.357142857142857 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nxfer
.ENDS $ comp_8 $

.SUBCKT comp_9
+ D G S
+ M=1
MXMO D G S VSS mn W=2.8U L=0.7U AD=2.38P AS=0P PD=4.5U PS=0U
NRD=0.178571428571429 NRS=0.178571428571429 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nmos
.ENDS $ comp_9 $

.SUBCKT comp_10
+ B D G S
+ M=1
MXMO D G S B mp W=4.55U L=1.465U AD=3.8675P AS=0P PD=6.25U
PS=0U NRD=0.10989010989011 NRS=0.10989010989011 M=1
* Cell /user/mldram1/HDRAM_sch_asim/pmosb
.ENDS $ comp_10 $

.SUBCKT comp_11
+ B D G S
+ M=1
MXMO D G S B mp W=0.9U L=0.35U AD=0.765P AS=0.765P PD=2.6U
PS=2.6U NRD=0.555555555555556 NRS=0.555555555555556 M=1
* Cell /user/mldram1/HDRAM_sch_asim/pxferb
.ENDS $ comp_11 $
```

```
* The following Changes are to allow for manipulation of
transistor sizes to model process variations.

.param
+wp1=2.8U
+lp1=0.7U
+wp2=2.8U
+lp2=0.7U
+wn1=2.8U
+ln1=0.7U
+wn2=2.8U
+ln2=0.7U

.SUBCKT comp_p1
+ B D G S
+ M=1
MXMO D G S B mp W=wp1 L=lp1 AD=2.38P AS=0P PD=4.5U PS=0U
NRD=0.178571428571429 NRS=0.178571428571429 M=1
* Cell /user/mldram1/sch_lib/comp_analog/pmosb
.ENDS $ comp_p1 $

.SUBCKT comp_p2
+ B D G S
+ M=1
MXMO D G S B mp W=wp2 L=lp2 AD=2.38P AS=0P PD=4.5U PS=0U
NRD=0.178571428571429 NRS=0.178571428571429 M=1
* Cell /user/mldram1/sch_lib/comp_analog/pmosb
.ENDS $ comp_p2 $

.SUBCKT comp_n1
+ D G S
+ M=1
MXMO D G S VSS mn W=wn1 L=ln1 AD=2.38P AS=0P PD=4.5U PS=0U
NRD=0.178571428571429 NRS=0.178571428571429 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nmos
.ENDS $ comp_n1 $

.SUBCKT comp_n2
+ D G S
+ M=1
MXMO D G S VSS mn W=wn2 L=ln2 AD=2.38P AS=0P PD=4.5U PS=0U
NRD=0.178571428571429 NRS=0.178571428571429 M=1
* Cell /user/mldram1/sch_lib/comp_analog/nmos
.ENDS $ comp_n2 $
```

```
.SUBCKT blsa_1
+ BL BL- DB DB- I PR- PS R S- VBLPI VDDI VSSI YS Z
+ M=1
XI_228  BITLINE I BL comp_3
+ M=1
XI_229  BITLINE- I BL- comp_3
+ M=1
XMPS  S- PS VSSI comp_4
+ M=1
XMPR  VDDI R PR- VDDI comp_5
+ M=1
************
* The Following Transistor is p1
XMSAP0  VDDI BITLINE BITLINE- R comp_p2
+ M=1

* The Following Transistor is p2:
XMSAP1  VDDI BITLINE- BITLINE R comp_p1
+ M=1

* The Following Transistor is n1:
XMSAN0  BITLINE BITLINE- S- comp_n1
+ M=1

* The Following Transistor is n2:
XMSAN1  BITLINE- BITLINE S- comp_n2
+ M=1
************
XI_11  BITLINE YS DB comp_3
+ M=1
XI_7  BITLINE- YS DB- comp_3
+ M=1
XMN_BL2  BITLINE- Z VBLPI comp_7
+ M=1
XMN_BL0  VBLPI Z BITLINE comp_7
+ M=1
XMN_BL1  BITLINE Z BITLINE- comp_8
+ M=1

* Cell /user/mldram1/sch_asim/blsa_1
.ENDS $ blsa_1 $

.SUBCKT x_connect
+ BLL BLL- BLR BLR- C C- EL ER PL PR VBLPI X X-
+ M=1
XMN_BL0  VBLPI PL BLL comp_7
```

```
+ M=1
XMN_BL1  BLL EL BLL- comp_8
+ M=1
XMN_BL2  BLL- PL VBLPI comp_7
+ M=1
XMN_BL4  BLR ER BLR- comp_8
+ M=1
XMN_BL3  VBLPI PR BLR comp_7
+ M=1
XMN_BL5  BLR- PR VBLPI comp_7
+ M=1
XI_22  BLL X BLR- comp_3
+ M=1
XI_16  BLL- X- BLR comp_3
+ M=1
XI_1  BLL C BLR comp_3
+ M=1
XI_9  BLL- C- BLR- comp_3
+ M=1
* Cell /user/mldram1/sch_asim/x_connect
.ENDS $ x_connect $

.SUBCKT cell
+ BL SUB WL
+ M=1
XI_224  SUB SUB NODE SUB comp_10
+ M=1
XI_1656  SUB NODE WL BL comp_11
+ M=1
* Cell /user/mldram1/schematic/cell
.ENDS $ cell $

.SUBCKT blsa_1_asim
+ BL[0] BL[1] BL[2] BL[3] BL-[0] BL-[1] BL-[2] BL-[3] DB[0]
DB[1] DB[2] DB[3] DB-[0] DB-[1] DB-[2] DB-[3] I_I PR-_I PS_I
R_I S-_I
+ VBLPI_I VDDI_I VSSI_I YS_I Z_I
+ M=1
XBL_SA2  BL[2] BL-[2] DB[2] DB-[2] I PR- PS R S- VBLPI VDDI
VSSI YS_I Z blsa_1
+ M=1
XBL_SA0  BL[0] BL-[0] DB[0] DB-[0] I PR- PS R S- VBLPI VDDI
VSSI YS_I Z blsa_1
+ M=1
XI_525  PS PS_I comp_0
+ M=1
```

```
XI_524  PR- PR-_I comp_0
+ M=1
RXI_305 S-_I S- 17.5K m=1
RXI_304 R_I R 17.5K m=1
XBL_SA3  BL[3] BL-[3] DB[3] DB-[3] I PR- PS R S- VBLPI VDDI
VSSI YS_I Z blsa_1
+ M=1
RXI_309 VSSI_I VSSI 17.5K m=1
RXI_310 VDDI_I VDDI 17.5K m=1
XBL_SA1  BL[1] BL-[1] DB[1] DB-[1] I PR- PS R S- VBLPI VDDI
VSSI YS_I Z blsa_1
+ M=1
RXI_300 VBLPI_I VBLPI 17.5K m=1
XI_527  I I_I comp_0
+ M=1
XI_526  Z Z_I comp_0
+ M=1
* Cell /user/mldram1/sch_asim/blsa_1_asim
.ENDS $ blsa_1_asim $

.SUBCKT x_con_asim
+ BLL[0] BLL[1] BLL[2] BLL[3] BLL-[0] BLL-[1] BLL-[2] BLL-
[3] BLR[0] BLR[1] BLR[2] BLR[3] BLR-[0] BLR-[1] BLR-[2] BLR-
[3] C-_I C_I
+ EL_I ER_I PL_I PR_I VBLPI_I X-_I X_I
+ M=1
XI_312  VBLPI VBLPI_I comp_0
+ M=1
XI_311  PR PR_I comp_0
+ M=1
XI_310  ER ER_I comp_0
+ M=1
XI_309  C- C-_I comp_0
+ M=1
XI_347  EL EL_I comp_0
+ M=1
XI_259  PL PL_I comp_0
+ M=1
XI_4  BLL[3] BLL-[3] BLR[3] BLR-[3] C C- EL ER PL PR VBLPI X
X- x_connect
XI_3  BLL[2] BLL-[2] BLR[2] BLR-[2] C C- EL ER PL PR VBLPI X
X- x_connect
XI_1  BLL[0] BLL-[0] BLR[0] BLR-[0] C C- EL ER PL PR VBLPI X
X- x_connect
XI_2  BLL[1] BLL-[1] BLR[1] BLR-[1] C C- EL ER PL PR VBLPI X
X- x_connect
```

```
XI_350  X X_I comp_0
+ M=1
XI_349  C C_I comp_0
+ M=1
XI_348  X- X-_I comp_0
+ M=1
* Cell /user/mldram1/sch_asim/x_con_asim
.ENDS $ x_con_asim $

.SUBCKT comp_1
+ BL_IN BL_IN- BL_IN_N BL_OUT BL_OUT- BL_OUT_N SUB WL[0]
WL[1] WL[2] WL[3]
+ M=1
CXI_1098 BL_IN VSS 3.168F m=1
RXI_1103 BL_IN BL_OUT 7.74  m=1
XCELL2  BL_OUT SUB WL[2] cell
+ M=3
CXI_1520 BL_OUT- VSS 3.168F m=1
RXI_1519 BL_IN- BL_OUT- 7.74  m=1
CXI_1102 BL_OUT VSS 3.168F m=1
XS_CELL  BL_OUT- SUB WL[0] cell
+ M=3
CXI_1522 BL_IN- VSS 3.168F m=1
XCELL3  BL_IN- SUB WL[3] cell
+ M=3
CXI_1513 BL_IN- BL_IN 1.122F m=1
CXI_1516 BL_OUT- BL_OUT_N 1.92F m=1
CXI_1517 BL_IN- BL_IN_N 1.92F m=1
CXI_1514 BL_OUT- BL_OUT 1.122F m=1
XCELL1  BL_IN SUB WL[1] cell
+ M=3
* Cell /user/mldram1/sch_asim/blm_nwl
.ENDS $ comp_1 $

.SUBCKT comp_2
+ BL_IN BL_IN- BL_IN_N BL_OUT BL_OUT- BL_OUT_N SUB WL[0]
WL[1] WL[2] WL[3]
+ M=1
CXI_1098 BL_IN VSS 1.056F m=1
RXI_1103 BL_IN BL_OUT 2.58  m=1
XCELL2  BL_OUT SUB WL[2] cell
+ M=1
CXI_1520 BL_OUT- VSS 1.056F m=1
RXI_1519 BL_IN- BL_OUT- 2.58  m=1
CXI_1102 BL_OUT VSS 1.056F m=1
XS_CELL  BL_OUT- SUB WL[0] cell
```

```
+ M=1
CXI_1522 BL_IN- VSS 1.056F m=1
XCELL3  BL_IN- SUB WL[3] cell
+ M=1
CXI_1513 BL_IN- BL_IN 0.374F m=1
CXI_1516 BL_OUT- BL_OUT_N 0.64F m=1
CXI_1517 BL_IN- BL_IN_N 0.64F m=1
CXI_1514 BL_OUT- BL_OUT 0.374F m=1
XCELL1  BL_IN SUB WL[1] cell
+ M=1
* Cell /user/mldram1/sch_asim/blm_nwl
.ENDS $ comp_2 $

.SUBCKT blmx4_16WL
+ BL0[0] BL0[1] BL0-[0] BL0-[1] BL1[0] BL1[1] BL1-[0] BL1-
[1] BL2[0] BL2[1] BL2-[0] BL2-[1] BL3[0] BL3[1] BL3-[0] BL3-
[1] SUB VDDI
+ WL_L
+ M=1
XI_4  BL3[1] BL3-[1] VSS N_19 N_23 VSS SUB VDDI VDDI VDDI
VDDI comp_1
+ M=1
XS_BLM_3  N_19 N_23 VSS BL3[0] BL3-[0] VSS SUB WL_L VDDI VDDI
VDDI comp_2
+ M=1
XI_3  BL2[1] BL2-[1] BL3[1] N_15 N_18 N_19 SUB VDDI VDDI VDDI
VDDI comp_1
+ M=1
XS_BLM_2  N_15 N_18 N_19 BL2[0] BL2-[0] BL3[0] SUB WL_L VDDI
VDDI VDDI comp_2
+ M=1
XI_2  BL1[1] BL1-[1] BL2[1] N_7 N_11 N_15 SUB VDDI VDDI VDDI
VDDI comp_1
+ M=1
XS_BLM_1  N_7 N_11 N_15 BL1[0] BL1-[0] BL2[0] SUB WL_L VDDI
VDDI VDDI comp_2
+ M=1
XI_1  BL0[1] BL0-[1] BL1[1] N_2 N_6 N_7 SUB VDDI VDDI VDDI
VDDI comp_1
+ M=1
XS_BLM_0  N_2 N_6 N_7 BL0[0] BL0-[0] BL1[0] SUB WL_L VDDI
VDDI VDDI comp_2
+ M=1
* Cell /user/mldram1/sch_asim/blmx4_16WL
.ENDS $ blmx4_16WL $
```

```
.SUBCKT blmx4_4WL
+ BL0[0] BL0[1] BL0-[0] BL0-[1] BL1[0] BL1[1] BL1-[0] BL1-
[1] BL2[0] BL2[1] BL2-[0] BL2-[1] BL3[0] BL3[1] BL3-[0] BL3-
[1] SUB WL0
+ WL1 WL2 WL3
+ M=1
XBLM_NWL_0  BL0[1] BL0-[1] BL1[1] BL0[0] BL0-[0] BL1[0] SUB
WL0 WL1 WL2 WL3 comp_2
+ M=1
XBLM_NWL_1  BL1[1] BL1-[1] BL2[1] BL1[0] BL1-[0] BL2[0] SUB
WL0 WL1 WL2 WL3 comp_2
+ M=1
XBLM_NWL_2  BL2[1] BL2-[1] BL3[1] BL2[0] BL2-[0] BL3[0] SUB
WL0 WL1 WL2 WL3 comp_2
+ M=1
XBLM_NWL_3  BL3[1] BL3-[1] VSS BL3[0] BL3-[0] VSS SUB WL0 WL1
WL2 WL3 comp_2
+ M=1
* Cell /user/mldram1/sch_asim/blmx4_4WL
.ENDS $ blmx4_4WL $

.SUBCKT blmx4
+ BL0_FSA BL0_FSA- BL0_NSA BL0_NSA- BL1_FSA BL1_FSA- BL1_NSA
BL1_NSA- BL2_FSA BL2_FSA- BL2_NSA BL2_NSA- BL3_FSA BL3_FSA-
BL3_NSA
+ BL3_NSA- DL DL- SUB VDDI WL[63]
+ M=1
XBLMX4_16WL_1   N_823 N_794 N_824 N_790 N_822 N_786 N_821
N_781 N_819 N_776 N_820 N_771 N_818 N_766 N_817 N_761 SUB
VDDI VDDI
+ blmx4_16WL
+ M=1
XS_BLMX4_16WL  BL0_FSA N_824 BL0_FSA- N_823 BL1_FSA- N_822
BL1_FSA N_821 BL2_FSA N_820 BL2_FSA- N_819 BL3_FSA- N_818
BL3_FSA N_817
+ SUB VDDI WL[63] blmx4_16WL
+ M=1
XBLMX4_16WL_3   N_791 N_796 N_795 N_792 N_787 N_816 N_782
N_2031 N_772 N_815 N_777 N_814 N_767 N_813 N_762 N_2032 SUB
VDDI VDDI
+ blmx4_16WL
+ M=1
XBLMX4_16WL_2   N_794 N_795 N_790 N_791 N_781 N_787 N_786
N_782 N_776 N_777 N_771 N_772 N_761 N_767 N_766 N_762 SUB
VDDI VDDI
+ blmx4_16WL
```

```
+ M=1
XBLMX4_4WL_0   N_796 BL0_NSA N_792 BL0_NSA- N_2031 BL1_NSA
N_816 BL1_NSA- N_815 BL2_NSA N_814 BL2_NSA- N_2032 BL3_NSA
N_813 BL3_NSA-
+ SUB DL- VDDI DL VDDI blmx4_4WL
+ M=1
* Cell /user/mldram1/sch_asim/blmx4
.ENDS $ blmx4 $

$$.SUBCKT array_asim
$$+ DUMMY_PIN
$$ + M=1
XBLMX4_L     BITLINEL0 BITLINEL0- N_467 N_739 BITLINTL1-
BITLINEL1 N_740 N_741 BITLINEL2 BITLINEL2- N_742 N_743
BITLINEL3- BITLINEL3
+ N_744 N_745 DLL DLL- VCP VDD WLL[63] blmx4
+ M=1
XBLMX4_R     BITLINER0 BITLINER0- N_730 N_731 BITLINER1-
BITLINER1 N_732 N_733 BITLINER2 BITLINER2- N_734 N_735
BITLINER3- BITLINER3
+ N_736 N_737 DLR DLR- VCP VDD WLR[63] blmx4
+ M=1
XBLSA_1_ASIM_R  N_730 N_732 N_734 N_736 N_731 N_733 N_735
N_737 DBR[0] DBR[1] DBR[2] DBR[3] DBR-[0] DBR-[1] DBR-[2]
DBR-[3] IR PRR-
+ PSR RR SR- VBLP VDD VSS YSR ZR blsa_1_asim
XI_410   BITLINEL0 BITLINEL1 BITLINEL2 BITLINEL3 BITLINEL0-
BITLINTL1-   BITLINEL2-   BITLINEL3-   BITLINER0   BITLINER1
BITLINER2 BITLINER3
+ BITLINER0- BITLINER1- BITLINER2- BITLINER3- C- C EL ER PL
PR VBLP X- X x_con_asim
XBLSA_1_ASIM_L  N_467 N_740 N_742 N_744 N_739 N_741 N_743
N_745 DBL[0] DBL[1] DBL[2] DBL[3] DBL-[0] DBL-[1] DBL-[2]
DBL-[3] IL PRL-
+ PSL RL SL- VBLP VDD VSS YSL ZL blsa_1_asim
* Cell /tmp_mnt/user/mldram1/sch_asim/array_asim
$$.ENDS $ array_asim $
```

A.2.2. The Stimulus Template File

```
*Stimulus file for array simulation
*File    name:    /user/mldram1/sch_asim/array_asim/hspice/
array_asim.stim

.option probe
```

```
$$$
```

```
+        t17      = 130n
+        t17d     = 133n
+        t18      = 140n
+        t18d     = 143n
+        t19      = 150n
+        t19d     = 153n
+        t20      = 170n
+        t20d     = 173n
+        t21      = 195n
+        t21d     = 198n
+        t22      = 200n
+        t22d     = 203n
+        t23      = 210n
+        t23d     = 213n
+        t24      = 215n
+        t24d     = 218n
+        t25      = 220n
+        t25d     = 223n
+tmeas= 229n
+        t26      = 230n

$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$

* DC stimulus

VDD VDD 0 DC PVDD
VSS VSS 0 DC PVSS
VCP VCP 0 DC PVCP
VBLP VBLP 0 DC PVBLP



* BLSA_1_ASIM_L (Left) Stimulus

VPRL- PRL- 0 PWL(t0 3.3 t7 3.3 t7d 0 t24 0 t24d 3.3 t26 3.3 R
t0)
VPSL PSL 0 PWL(t0 0 t7 0 t7d 3.3 t24 3.3 t24d 0 t26 0 R t0)
VRL RL 0 PWL(t0 0 t7 0 t7d 3.3 t24 3.3 t24d 0 t26 0 R t0)
VSL- SL- 0 PWL(t0 3.3 t7 3.3 t7d 0 t24 0 t24d 3.3 t26 3.3 R
t0)
VIL IL 0 PWL(t0 0 t6 0 t6d 5.5 t8 5.5 t8d 0 t18 0 t18d 5.5
t21 5.5 t21d 0 t26 0 R t0)
VZL ZL 0 PWL(t0 5.5 t4 5.5 t4d 0 t25 0 t25d 5.5 t26 5.5 R t0)
VYSL YSL 0 0v
```

```
* BLMX$4_L (left) stimulus

VWLi WLL[63] 0 PWL(t0 pvdd t3 pvdd t3d pvpp t9 pvpp t9d pvdd
t13 pvdd t13d pvpp t23 pvpp t23d pvdd t26 pvdd R t0)
VDLL DLL 0 pvpp
VDLL- DLL- 0 PWL(t0 pvpp t2 pvpp t2d pvdd t10 pvdd t10d pvpp
t12 pvpp t12d pvdd t25 pvdd t25d pvpp t26 pvpp R t0)

* X_CON_ASIM stimulus

VPR PR 0 PWL(t0 5.5 t1 5.5 t1d 0 t25 0 t25d 5.5 t26 5.5 R t0)
VER ER 0 PWL(t0 5.5 t1 5.5 t1d 0 t22 0 t22d 5.5 t26 5.5 R t0)
VX- X- 0 PWL(t0 0 t20 0 t20d 5.5 t25 5.5 t25d 0 t26 0 R t0)
VC C 0 PWL(t0 0 t13 0 t13d 5.5 t15 5.5 t15d 0 t26 0 R t0)
VX X 0 0v
VC- C- 0 PWL(t0 0 t4 0 t4d 5.5 t5 5.5 t5d 0 t26 0 R t0)
VEL EL 0 PWL(t0 5.5 t1 5.5 t1d 0 t10 0 t10d 5.5 t16 5.5 t16d
0 t25 0 t25d 5.5 t26 5.5 R t0)
VPL PL 0 PWL(t0 5.5 t1 5.5 t1d 0 t10 0 t10d 5.5 t11 5.5 t11d
0 t25 0 t25d 5.5 t26 5.5 R t0)

* BLMX4_R (right) stimulus

VWRi WLR[63] 0 pvdd
VDLR DLR 0 pvpp
VDLR- DLR- 0 pvpp

* BLSA_1_ASIM_R (Right) Stimulus

VPRR- PRR- 0 PWL(t0 3.3 t17 3.3 t17d 0 t24 0 t24d 3.3 t26 3.3
R t0)
VPSR PSR 0 PWL(t0 0 t17 0 t17d 3.3 t24 3.3 t24d 0 R t26 t0)
VSR- SR- 0 PWL(t0 3.3 t17 3.3 t17d 0 t24 0 t24d 3.3 t26 3.3 R
t0)
VRR RR 0 PWL(t0 0 t17 0 t17d 3.3 t24 3.3 t24d 0 R t26 t0)
VIR IR 0 PWL(t0 0 t16 0 t16d 5.5 t19 5.5 t19d 0 t26 0 R t0)
VZR ZR 0 PWL(t0 5.5 t14 5.5 t14d 0 t25 0 t25d 5.5 t26 5.5 R
t0)
VYSR YSR 0 0v

* DATA BUS stimulus

VDBL0 DBL[0] 0 0V
VDBL0- DBL-[0] 0 0V
VDBL1 DBL[1] 0 0V
```

```
VDBL1- DBL-[1] 0 0V
VDBL2 DBL[2] 0 0V
VDBL2- DBL-[2] 0 0V
VDBL3 DBL[3] 0 0V
VDBL3- DBL-[3] 0 0V
VDBR0 DBR[0] 0 0V
VDBR0- DBR-[0] 0 0V
VDBR1 DBR[1] 0 0V
VDBR1- DBR-[1] 0 0V
VDBR2 DBR[2] 0 0V
VDBR2- DBR-[2] 0 0V
VDBR3 DBR[3] 0 0V
VDBR3- DBR-[3] 0 0V

.TRAN 23.0n t26

* The cells connected to WLL(63) of the cct /user/mldram1/
sch_asim/array_asim
* xblmx4_1.xs_blmx4_16wl.xs_blm_3.xs_cell.node   =   Level3
* xblmx4_1.xs_blmx4_16wl.xs_blm_2.xs_cell.node   =   Level2
* xblmx4_1.xs_blmx4_16wl.xs_blm_1.xs_cell.node   =   Level1
* xblmx4_1.xs_blmx4_16wl.xs_blm_0.xs_cell.node   =   Level0


.PRINT TRAN
+ V(xblmx4_1.xs_blmx4_16wl.xs_blm_3.xs_cell.node)
+ V(xblmx4_1.xs_blmx4_16wl.xs_blm_2.xs_cell.node)
+ V(xblmx4_1.xs_blmx4_16wl.xs_blm_1.xs_cell.node)
+ V(xblmx4_1.xs_blmx4_16wl.xs_blm_0.xs_cell.node)

.IC
+    xblmx4_1.xs_blmx4_16wl.xs_blm_3.xs_cell.node          =
initial_voltage
+    xblmx4_1.xs_blmx4_16wl.xs_blm_2.xs_cell.node          =
initial_voltage
+    xblmx4_1.xs_blmx4_16wl.xs_blm_1.xs_cell.node          =
initial_voltage
+    xblmx4_1.xs_blmx4_16wl.xs_blm_0.xs_cell.node          =
initial_voltage

*****************************************************
.param ln1 = %LENGTH%U
.param initial_voltage = %INIT_VOLTAGE%

.END
```

APPENDIX B: TEST VECTOR GENERATION PROGRAM

The following program was used to generate vectors for testing the first MLDRAM test chip in DRAM mode. The IMS setup file is appended at the end of the program.

C++ Test Vector Generation Program for 5n March Test

```
// testgen.cc
// Test pattern generator for IMS tester

#include <iostream.h>
#define ROWSIZE 64
#define COLSIZE 8
#define VAL     0
#define VALBAR  1
#define ADDRSIZE 6
#define VECTORSIZE 19


/* output format:
vcp, vblp, ,clk, addr5, addr4, addr3, addr2, addr1, addr0
xclk, xden, sx1, xs0, z, sense,
din, yden, write, dout

The test pattern writes VAL to the entire array, then,
starting from the top left, it does the following:
read VAL, write VALBAR, read VALBAR

*/

// declare global line# counter to make life really easy
int vectornumber=0;

// set output order
const int vcp=0 , vblp=1, clk=2 ;
int addr5=3;
// int addr4=4, addr3=5, addr2=6, addr1=7, addr0=8;
const int xclk=9, xden=10, sx1=11, sx0=12, z=13, sense=14;
const int din=15, yden=16, wri=17, dout=18;

// declare global output vector
int vector[VECTORSIZE];


int *binary(int x, int  *vector){
```

```
//  output binary form of int
    int i=6;
    for(i=ADDRSIZE-1;i>=0;i--){
       vector[i]=(x%2);
       x/=2;
    }
    return vector;
}


int scramblerow(int x){
// scrambling function for row

    if ((x%4)==1)
       return (x+1);
    else if ((x%4==2))
       return (x-1);
    else
       return (x);
}

int scramblecol(int x){
    x+=8;
    return (x);
}

void printvector(int *vector){
    int i;

    cout << vectornumber++ << ":\t";
    for (i=0;i<VECTORSIZE;i++){
       if (vector[i]==-1)
          cout << "x";
       else
          cout << vector[i];
                                                                    if
((i==2)||(i==8)||(i==10)||(i==12)||(i==14)||(i==17))
          cout << " ";
    }
    cout << endl;
}

void cycle(int row, int col, int rw, int data){

// define address pointer
int *addr = vector+addr5;
```

```
cout << "rem ";
if (rw)  cout << "Write "; else cout << "Read ";
if (data)  cout << "HIGH "; else cout << "LOW ";
if (rw)  cout << "to   ("; else cout << "from (";
cout << row << "," << col << ")\n";

vector[vcp]=1;
vector[vblp]=1;
vector[clk]=1;
vector[xclk]=1;
vector[xden]=0;
vector[sx0]=0;
vector[sx1]=0;
vector[z]=1;
vector[sense]=0;
vector[yden]=0;
vector[wri]=0;
vector[dout]=-1;
binary(scramblerow(row),addr);

printvector(vector);

// set column address, set data-in, enable column decode,
disable precharge
binary(scramblecol(col), addr);


vector[clk]=0;
vector[xclk]=0;
vector[xden]=1;
if((scramblerow(row)%2)==0)
  vector[sx0]=1;
else
  vector[sx1]=1;
vector[z]=0;
vector[sense]=1;
vector[din]=(scramblerow(row)%2)^data;
vector[yden]=1;
vector[wri]=rw;
vector[dout]=(scramblerow(row)%2)^data;

printvector(vector);

}
```

```
int main(){

   int row, col;
   int zero=0;
   int one=1;
   int rw=1;
   int read=0;

   cout << "cvt\nRestore_Defaults\nUnits=XXXXXXns\n\n";
   cerr << "Don't forget to specify units\n";

   cout << "rem Initial state:\n";
   cout << "0:\tzzz zzzzzz zz zz zz zzz x\n\n";
   vectornumber++;

   cout << "rem Optimal march test for fault detection
(5n)\n\n";

  for(row=0;row<ROWSIZE;row++){
      for(col=0;col<COLSIZE;col++){
         cycle(row, col, rw  , zero);
         cout << endl;
      }
   }
// wait 13000 clocks
//vectornumber+=5000;
 for(row=0;row<ROWSIZE;row++){
      for(col=0;col<COLSIZE;col++){
         cycle(row, col, read, zero);
         cycle(row, col, rw  , one );
         cout << endl;
      }
   }
//vectornumber+=5000;
 for(row=0;row<ROWSIZE;row++){
      for(col=0;col<COLSIZE;col++){
         cycle(row, col, read, one );
         cycle(row, col, rw  , zero);
         cout << endl;
      }
   }
cerr << vectornumber-1 << " patterns\n";
}
```

IMS Setup File for Optimal Access Time

```
Rem ID Logic Master XL-60 X4.90F   Options: Matrix, 1 Hard
disk
Rem [1] "Fast 35.7ns setup"
Rem [2] ""
Init
SRQ 7
Clk Internal,38.00ns
Event 0=User0,Off
Event 1=User1,Off
Event 2=STM-2,Off
Event 3=Error,On
Socket "XL DIP Auto", 64, "XL DIP Auto"
Calibrate Fixture Offset 0ns
Config 0,"XL-60 Control","16K","Stimulation Control"
Config 4,"XL-60 Split Data","16K","16 Force, 16 Compare
Channels"
Config 5,"XL-60 Split Data","16K","16 Force, 16 Compare
Channels"
Config 6,"XL-60 Split Data","16K","16 Force, 16 Compare
Channels"
Config 8,"XL Timing","","System Timing, 1 Power Supply"
Config 15,"XL PMU","","DC Parametric Measurement"

Testconditions #TXT
  Start All
  Last Vector Maintain
  Branch Vector Maintain
  PMU Off
  Autoranging Enabled
  Nominal Accuracy 0.1
Testconditions End

Equation Recomputation Manual
Equation End

Resource VCP=Force #TXT
  26, vcp
Resource End
Radix VCP=Bin
Polarity VCP=Positive
Lodrive VCP=0mV
Hidrive VCP=4.50V

Format VCP=NRZ
```

```
Resource VBLP=Force #TXT
  39, vblp
Resource End
Radix VBLP=Bin
Polarity VBLP=Positive
Lodrive VBLP=0mV
Hidrive VBLP=2.60V

Format VBLP=NRZ


Resource CLK=Force #TXT
  45, clk
Resource End
Radix CLK=Bin
Polarity CLK=Positive
Lodrive CLK=0mV
Hidrive CLK=3.30V

Format CLK=NRZ


Resource ADDR=Force #TXT
  14, addr5
  13, addr4
  15, addr3
  12, addr2
  11, addr1
  10, addr0
Resource End
Radix ADDR=Bin
Polarity ADDR=Positive
Lodrive ADDR=0mV
Hidrive ADDR=3.30V

Format ADDR=NRZ


Resource XCLK=Force #TXT
  53, xclk
Resource End
Radix XCLK=Bin
Polarity XCLK=Positive
Lodrive XCLK=0mV
```

```
Hidrive XCLK=3.30V

Format XCLK=DNRZ,20.0ns


Resource XDEN=Force #TXT
  50, xden
Resource End
Radix XDEN=Bin
Polarity XDEN=Positive
Lodrive XDEN=0mV
Hidrive XDEN=3.30V

Format XDEN=RZ,0ns,30.5ns


Resource SX=Force #TXT
  55, sx1
  54, sx0
Resource End
Radix SX=Bin
Polarity SX=Positive
Lodrive SX=0mV
Hidrive SX=3.30V

Format SX=RZI,6.5ns,29.0ns


Resource Z=Force #TXT
  49, z
Resource End
Radix Z=Bin
Polarity Z=Negative
Lodrive Z=0mV
Hidrive Z=3.30V

Format Z=NRZ


Resource SENSE=Force #TXT
  47, sense
Resource End
Radix SENSE=Bin
Polarity SENSE=Positive
Lodrive SENSE=0mV
Hidrive SENSE=3.30V
```

```
Format SENSE=RZ,21.0ns,12.3ns


Resource DIN=Force #TXT
  16, din
Resource End
Radix DIN=Bin
Polarity DIN=Negative
Lodrive DIN=0mV
Hidrive DIN=3.30V

Format DIN=NRZ


Resource YDEN=Force #TXT
  51, yden
Resource End
Radix YDEN=Bin
Polarity YDEN=Positive
Lodrive YDEN=0mV
Hidrive YDEN=3.30V

Format YDEN=RZ,28.6ns,7.1ns


Resource RW=Force #TXT
  52, rw
Resource End
Radix RW=Bin
Polarity RW=Positive
Lodrive RW=0mV
Hidrive RW=3.30V

Format RW=RZ,30.0ns,4.0ns


Resource DOUT=Compare #TXT
  17, dout
Resource End
Radix DOUT=Bin
Polarity DOUT=Positive
Threshold DOUT=1.65V

Sample DOUT=35.7ns
```

```
Resource VCC=Power,VA #TXT
  25, vdd
Resource End
Power VCC=3.300V,250mA,0ms,LOZ


Resource VSS=Power,GND #TXT
  44, vss
Resource End
Power VSS=0V,,,GND


Equation End
Equation Recomputation Manual

Waveform Display #TXT
Waveform End
Waveform Magnify=20
Waveform Markers Sequence="Off","Off"

Display Compare="Expect and Acquire"
Fail 641
Mask "1"
Acp Setup_Hold Binary #TXT
Acp Prop_Delay Binary #TXT
Acp End
Fail Acp Setup = 0, Hold = 0, Prop = 0, Analysis = 0


PAR End
PAR GLT #TXT
  Connect=
  Open=
PAR End

DCP VCP, None
DCC VCP, None
DCResistance VCP, 0m

DCP VBLP, None
DCC VBLP, None
DCResistance VBLP, 0m

DCP CLK, None
DCC CLK, None
```

```
DCResistance CLK, 0m

DCP ADDR, None
DCC ADDR, None
DCResistance ADDR, 0m

DCP XCLK, None
DCC XCLK, None
DCResistance XCLK, 0m

DCP XDEN, None
DCC XDEN, None
DCResistance XDEN, 0m

DCP SX, None
DCC SX, None
DCResistance SX, 0m

DCP Z, None
DCC Z, None
DCResistance Z, 0m

DCP SENSE, None
DCC SENSE, None
DCResistance SENSE, 0m

DCP DIN, None
DCC DIN, None
DCResistance DIN, 0m

DCP YDEN, None
DCC YDEN, None
DCResistance YDEN, 0m

DCP RW, None
DCC RW, None
DCResistance RW, 0m

DCP DOUT, None
DCC DOUT, None
DCResistance DOUT, 0m


MASK DCP NONE
FAIL DCP Failed = 0, Untested = 0, Passed = 0
End
```

APPENDIX C: CHARGE INJECTION ANALYSIS

C.1 Charge Injection Theory

When a transistor is turned on, some charge resides in the inversion region. The total gate-to-channel capacitance C_G is

$$C_G = WLC_{OX} \tag{C.1}$$

Generally, the control signals for multilevel DRAM are switched from high to low and vice-versa and so the transistors are either in linear operation or in cutoff. While in linear operation, we assume that there is ample time for the source and drain capacitances to equalize to the same potential. Thus when a switch is turned on, the next timing event is assumed to occur after the source and drain are equal in potential.

The digital switching operations used here have fixed high and low values. Let V_{onm} be the voltage used to turn transistor 'm' on. Let the charge Q_{im} injected into the equalized source-drain circuit node be represented as a function of the source voltage V_S. This can be written as:

$$Q_{im}(V_S) = \text{Charge injected into source and drain as a result of switch } m \text{ activation.} \tag{C.2}$$

In addition to the fixed gate voltage for linear operation, most operations in multilevel DRAM occur where the source and drain are near precharge, V_{BLP}. V_S is not always exactly V_{BLP} but it only varies slightly from V_{BLP} due to the small voltages used in sensing operations. Let C_B be the total parasitic capacitance for one sub-bitline and let it be linear and equal for all sub-bitlines. Let C_C be the cell capacitance and let it be linear and equal for all cells including dummy cells. For a typical choice of C_B and C_C the bitline voltages will not vary by more than 10%. Thus a new terminology can be introduced to replace Eq. C.2. Assuming that V_S is within 10% of V_{BLP}, the injected charge $Q_{im}(V_S)$

can be considered as a constant equal to $Q_{im}(V_{BLP})$. In the following discussion, $Q_{im}(V_{BLP})$ from Eq. C.2 will be denoted simply as Q_{im}.

$$Q_{im} = \text{Charge injected into source and drain when } V_S = V_{BLP} \tag{C.3}$$

For example, a rising transition on the E0 signal would cause the charge $+Q_{iE0}$ to be injected onto the source/drain node. Conversely a falling edge on the E0 signal would cause $-Q_{iE0}$ to be injected evenly onto the source and drain. As with turning a transistor on, $-Q_{im}(V_S)$ is expressed as in Eq. C.2 (and Eq. C.3) but V_S represents the initial source voltage rather than the final source voltage. In all cases, the injected charge is split evenly between source and drain when a transistor is turned off.

The theory applied to the circuit for charge sharing analysis is conservation of charge:

$$\sum_i C_i(t)V_i(t) = Q \quad \text{and} \quad \frac{dQ}{dt} = 0 \tag{C.4}$$

where the group of capacitances C_i, having a voltage V_i constitute a closed system. For this analysis the gate capacitance of the transistor switch is not considered for charge sharing operations. That is, while the gate capacitance causes charge to be injected due to the transition on its gate, the gate capacitance does not participate in charge sharing. This approximation is valid for most cases since for reasonable transistor sizes C_G is thirty to one hundred times smaller than the capacitance involved in charge sharing

C.2 Multilevel Restore

Table C-1 outlines the charge sharing steps to restore the multilevel data in the memory cell. For each step the voltage on each sub-bitline is calculated. This way the final restored cell voltage can be determined. Time is indexed by each event 'n' that affects the

final restored voltage. 'n' begins at a negative value since the index 0 is reserved for the beginning of a sense/restore cycle, as presented in Gillingham's paper.

TABLE C-1. Charge sharing events affecting the voltage that is restored

Time (n)	Event	Controls	affected nodes
-6	Initial conditions		
-5	Turn off I1 to isolate the sub-bitlines B10, B11	I1	B10, B11
-4	Connect X0 to charge sub-bitline B11	X0	B11
-3	Turn off I0 to isolate sub-bitlines	I0	B00, B11
-2	Turn on E1 to create final multilevel voltage	E1	B00, B11, B10
-1	Turn off wordline to complete restore	WL	MC, B00, B11, B10

For the analysis, let V_{SIGN} be the digital voltage of the sign bit. Possible values are V_{DD} and 0V; the complementary voltage is $V_{SIGN}^* = V_{DD} - V_{SIGN}$. Let V_{MAG} be the digital voltage of the magnitude bit; allowed values are V_{DD} and 0V; its complement is V_{MAG}^*. Let V_{DATA} be the multilevel data voltage. V_{DATA} is calculated from

$$V_{DATA} = \frac{2V_{SIGN} + V_{MAG}}{3} \tag{C.5}$$

Initially both the sign and magnitude bits are latched and the restore is about to begin. Both I0 and I1 are on and all cross connect signals are off. B01 is not used in restore so calculations of V_{B01} are omitted. The initial bitline conditions are:

$$V_{B00}(-6) = V_{SIGN} \tag{C.6}$$

$$V_{B10}(-6) = V_{MAG} \tag{C.7}$$

$$V_{B11}(-6) = V_{MAG}^* \tag{C.8}$$

Now disconnect the magnitude sense amplifier:

$$V_{B00}(-5) = V_{B00}(-6) \tag{C.9}$$

$$V_{B10}(-5) = V_{B10}(-6) + \frac{\left(-\frac{1}{2}\right)Q_{i11}(V_{MAG})}{C_B + C_C} \tag{C.10}$$

$$V_{B11}(-5) = V_{B11}(-6) + \frac{\left(-\frac{1}{2}\right)Q_{i11}(V_{MAG}^*)}{C_B + C_C} \tag{C.11}$$

Turn on X0 to charge B11 to the value of V_{SIGN}:

$$V_{B00}(-4) = V_{B00}(-5) \tag{C.12}$$

$$V_{B10}(-4) = V_{B10}(-5) \tag{C.13}$$

$$V_{1C}(-4) = V_{SIGN} \tag{C.14}$$

Disconnect the sign sense amplifier to isolate B00 and B11:

$$V_{B00}(-3) = V_{B00}(-4) + \frac{\left(-\frac{1}{2}\right)Q_{i10}(V_{SIGN})}{2C_B + 2C_C} \tag{C.15}$$

$$V_{B10}(-3) = V_{B10}(-4) \tag{C.16}$$

$$V_{B11}(-3) = V_{B11}(-4) + \frac{\left(-\frac{1}{2}\right)Q_{i10}(V_{SIGN})}{2C_B + 2C_C} \tag{C.17}$$

Now activate E1 to short the three sub-bitlines together

$$V_{B00}(-2) = \frac{(2C_B + 2C_C)V_{B00}(-3) + (C_B + C_C)V_{B10}(-3) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} \quad \text{(C.18)}$$

$$V_{B01}(-2) = \frac{(C_B + C_C)V_{B01}(-3) + (2C_B + 2C_C)V_{B11}(-3) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} \quad \text{(C.19)}$$

$$V_{B11}(-2) = \frac{(2C_B + 2C_C)V_{B11}(-3) + (C_B + C_C)V_{B10}(-3) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} \quad \text{(C.20)}$$

Finally, deactivate the wordline to capture the multilevel data in the cell:

$$V_{MC}(-1) = V_{B00}(-2) + \frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C} \quad \text{(C.21)}$$

Now the memory cell voltage can be found by solving Eq. C.5 through Eq. C.21:

$$V_{MC}(-1) = \frac{2V_{SIGN} + V_{MAG}}{3} + \frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C} + \frac{\left(-\frac{1}{2}\right)Q_{i10}(V_{SIGN}) + \left(-\frac{1}{2}\right)Q_{i11}(V_{MAG}) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} \quad \text{(C.22)}$$

The restored value contains voltage components due to charge injected during the restore process. All four charge injection terms are data-dependent. The memory cell voltage can be divided into signal and error components:

$$V_{MC}(-1) = V_{MC}(SIGNAL) + V_{MC}(ERROR) \quad \text{(C.23)}$$

$$V_{MC}(SIGNAL) = V_{DATA} = \frac{2V_{SIGN} + V_{MAG}}{3} \quad \text{(C.24)}$$

$$V_{MC}(ERROR) = \frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C} + \frac{\left(-\frac{1}{2}\right)Q_{i10}(V_{SIGN}) + \left(-\frac{1}{2}\right)Q_{i11}(V_{MAG}) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} \quad \text{(C.25)}$$

C.3 Sign Sensing

Table C-2 outlines the events in time (n) that affect the sign sensing.

TABLE C-2. Charge sharing events affecting sign sensing

Time (n)	Event	Controls	affected nodes
0	Initial conditions		All
1	Turn off precharge and equalize	E0, E1, P0, P1	All
2	Turn off dummy wordline	DB00	B00
3	Turn on wordline	WL	B00
4	Turn on C0	C0	B00, B10
5	Turn off C0	C0	B00, B10
6	Turn on I0	I0	B00, B01, SA00, SA01

Initially all bitlines are equalized and precharged. The precharge on B00 and B01 may be different from B10 and B11. Thus V_{BLP0} is the initial voltage of B00 and B01 while V_{BLP1} is the initial voltage of B10 and B11. For the analysis the voltages $V_{B00}(n)$, $V_{B01}(n)$,$V_{B10}(n)$, and $V_{B11}(n)$ will be evaluated for all n in Table C-1 and Table C-2.

$$V_{B00}(0) = V_{BLP0} \tag{C.26}$$

$$V_{B01}(0) = V_{BLP0} \tag{C.27}$$

$$V_{B10}(0) = V_{BLP1} \tag{C.28}$$

$$V_{B11}(0) = V_{BLP1} \tag{C.29}$$

Turn off precharge and equalize:

$$V_{B00}(1) = V_{B00}(0) + \frac{\left(-\frac{1}{2}\right)(Q_{iP0} + Q_{iE0})}{C_C + C_B} \tag{C.30}$$

$$V_{B01}(1) = V_{B01}(0) + \frac{\left(-\frac{1}{2}\right)(Q_{iP0} + Q_{iE0})}{C_C + C_B} \tag{C.31}$$

$$V_{B10}(1) = V_{B10}(0) + \frac{\left(-\frac{1}{2}\right)(Q_{iP1} + Q_{iE1})}{C_C + C_B} \tag{C.32}$$

$$V_{B11}(1) = V_{B11}(0) + \frac{\left(-\frac{1}{2}\right)(Q_{iP1} + Q_{iE1})}{C_C + C_B} \tag{C.33}$$

Turn off dummy wordline:

$$V_{B00}(2) = V_{B00}(1) + \frac{\left(-\frac{1}{2}\right)(Q_{iD0T})}{C_B} \tag{C.34}$$

$$V_{B01}(2) = V_{B01}(1) \tag{C.35}$$

$$V_{B10}(2) = V_{B10}(1) \tag{C.36}$$

$$V_{B11}(2) = V_{B11}(1) \tag{C.37}$$

Turn on wordline:

$$V_{B00}(3) = \frac{C_B V_{B00}(2) + C_C V_{MC}(-1) + Q_{iWL}}{C_B + C_C} \tag{C.38}$$

$$V_{B01}(3) = V_{B01}(2) \tag{C.39}$$

$$V_{B10}(3) = V_{B10}(2) \tag{C.40}$$

$$V_{B11}(3) = V_{B11}(2) \tag{C.41}$$

Turn on C0 to share data from B00 to B10:

$$V_{B00}(4) = \frac{(C_B + C_C)(V_{B00}(3) + V_{B10}(3)) + Q_{iC0}}{2C_B + 2C_C} \tag{C.42}$$

$$V_{B01}(4) = V_{B01}(3) \tag{C.43}$$

$$V_{B10}(4) = \frac{(C_B + C_C)(V_{B00}(3) + V_{B10}(3)) + Q_{iC0}}{2C_B + 2C_C} \tag{C.44}$$

$$V_{B11}(4) = V_{B11}(3) \tag{C.45}$$

Turn off C0 to prepare for sign sensing:

$$V_{B00}(5) = V_{B00}(4) + \frac{\left(-\frac{1}{2}\right)(Q_{iC0})}{C_B + C_C} \tag{C.46}$$

$$V_{B01}(5) = V_{B01}(4) \tag{C.47}$$

$$V_{B10}(5) = V_{B10}(4) + \frac{\left(-\frac{1}{2}\right)(Q_{iC0})}{C_B + C_C} \tag{C.48}$$

$$V_{B11}(5) = V_{B11}(4) \tag{C.49}$$

Turn on I0 to share the bitlines into the sense amplifier:

$$V_{B00}(6) = \frac{(C_C + C_B)V_{B00}(5) + C_{SA}V_{SA00}(5) + Q_{i10}}{C_C + C_B + C_{SA}} \tag{C.50}$$

$$V_{B01}(6) = \frac{(C_C + C_B)V_{B01}(5) + C_{SA}V_{SA01}(5) + Q_{i10}}{C_C + C_B + C_{SA}} \tag{C.51}$$

$$V_{B10}(6) = V_{B10}(5) \tag{C.52}$$

$$V_{B11}(6) = V_{B11}(5) \tag{C.53}$$

It is assumed that $V_{SA00}(5) = V_{SA01}(5)$. At this point the sign bit is sensed by connecting R0 and S0 to V_{DD} and V_{SS}. The differential mode signal to be sensed is V_{SD} and it is calculated using the following equation.

$$V_{SD} = V_{B00}(6) - V_{B01}(6) \tag{C.54}$$

which reduces to

$$V_{SD} = \left(\frac{(C_C + C_B)}{C_C + C_B + C_{SA}}\right)(V_{B10}(5) - V_{B11}(5)) \tag{C.55}$$

Clearly the signal V_{SD} is only the differential $V_{B00}(5)$ - $V_{B01}(5)$ diluted by the capacitance of the sense amplifier. The injected charge from I0 is cancelled in the differential mode. Eq. C.55 can be solved with some effort by solving Eq. C.26 through to Eq. C.54. For tractability, the voltages are calculated at n=5 first.

$$V_{B00}(5) = \tag{C.56}$$

$$\left(\frac{C_B}{C_B + C_C}\right)\left(\frac{V_{B00}(0)}{2} + \frac{\left(-\frac{1}{2}\right)(Q_{iP0})}{2C_B + 2C_C} + \frac{\left(-\frac{1}{2}\right)(Q_{iE0})}{2C_B + 2C_C}\right) + \frac{Q_{iWL} + \left(-\frac{1}{2}\right)(Q_{iDoT})}{2C_B + 2C_C} + \frac{C_C V_{MC}(-1)}{2C_B + 2C_C} +$$

$$\frac{V_{B01}(0)}{2} + \frac{\left(-\frac{1}{2}\right)(Q_{iP1})}{2C_B + 2C_C} + \frac{\left(-\frac{1}{2}\right)(Q_{iE1})}{2C_B + 2C_C}$$

Assuming that the charge injected by P0 is the same as that of P1, Q_{iP} is defined as

$$Q_{iP} = Q_{iP0} = Q_{iP1} \tag{C.57}$$

And similarly,

$$Q_{iE} = Q_{iE0} = Q_{iE1} \tag{C.58}$$

$$Q_{iW} = Q_{iWL} = Q_{iD0T} \tag{C.59}$$

So Eq. C.56 reduces to

$$V_{B00}(5) = \tag{C.60}$$

$$\left(\frac{C_B}{C_B + C_C}\right)\left(\frac{V_{0T}(0)}{2}\right) + \frac{V_{B01}(0)}{2} + \frac{C_C V_{MC}(-1)}{2C_B + 2C_C} + \frac{\frac{1}{2}Q_{iW}}{2C_B + 2C_C} - \frac{\left(\frac{2C_B + C_C}{2C_B + 2C_C}\right)(Q_{iP} + Q_{iE1})}{2C_B + 2C_C}$$

Before continuing to derive V_{SD}, it is worth noting that $V_{B00}(5)$ is made up of three distinct parts. The first two terms of Eq. C.60 represent the precharge value of sub-bitline B00. The scaling of $V_{B00}(0)$ by a capacitor ratio comes about because part of the precharge $V_{B00}(0)$ was lost when DB00 was deactivated. Simply put, a portion of precharge charge resides in the dummy cell. This effect is crucial to sensing as will become apparent when examining the final expression for V_{SD}. The third term is, of course, the memory cell data charge-shared from the cell onto two sub-bitlines and two cells. The fourth term is the injected wordline charge which did not get canceled when the dummy cell was turned off. The last two terms of Eq. C.60 represent the effect of charge injection by P0, P1, E0 and E1 onto sub-bitline B00. It expected that the coefficient of each would be unity divided by the total capacitance, $2C_B+2C_C$ but this is not the case since some of the charge Q_{iP0} and Q_{iE0} was lost forever in the dummy cell at time n=2.

Continuing, the other voltage at n=5 is

$$V_{B10}(5) = V_{B10}(0) + \frac{-Q_{iP0} - Q_{iE0}}{2C_C + 2C_B} \tag{C.61}$$

So V_{SD} is

$$V_{SD} = \left(\frac{(C_C + C_B)}{C_C + C_B + C_{SA}}\right)\Bigg(\tag{C.62}$$

$$\left(\frac{C_B}{C_B + C_C}\right)\left(\frac{V_{BLP0}}{2}\right) + \frac{V_{BLP1}}{2} - V_{BLP0} + \frac{C_C V_{DATA}}{2C_B + 2C_C}$$

$$+ \frac{C_C V_{MC}(ERROR)}{2C_B + 2C_C} + \frac{\frac{1}{2}Q_{iW}}{2C_B + 2C_C} + \frac{C_C(Q_{iP} + Q_{iE})}{(2C_B + 2C_C)^2}\Bigg)$$

If the precharge levels are both equal to V_{BLP} then

$$V_{SD} = \left(\frac{1}{C_C + C_B + C_{SA}}\right)\Bigg(\frac{C_C}{2}(V_{DATA} - V_{BLP} + V_{MC}(ERROR)) + \tag{C.63}$$

$$\frac{1}{4}Q_{iW} + \frac{\left(\frac{1}{2}\right)C_C}{2C_B + 2C_C}(Q_{iP} + Q_{iE})\Bigg)$$

This equation can be split into signal and error components:

$$V_{SD} = V_{SD}(SIGNAL) + V_{SD}(ERROR) \tag{C.64}$$

$$V_{SD}(SIGNAL) = \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{V_{DATA} - V_{BLP}}{2}\right) \tag{C.65}$$

$$V_{SD}(ERROR) = -\left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{4}Q_{iW} + \frac{C_C(Q_{iP} + Q_{iE})}{4C_B + 4C_C} + \frac{C_C}{2}V_{MC}(ERROR)\right) \tag{C.66}$$

To eliminate V_{SD}(ERROR) solve Eq. C.66. 66 for V_{SD}(ERROR) = 0.

$$Q_{iW} + 2C_C V_{MC}(ERROR) + \frac{C_C(Q_{iP} + Q_{iE})}{C_B + C_C} = 0 \tag{C.67}$$

C.4 Magnitude Sensing

Table C-3 outlines the events in time (n) that affect the magnitude sensing.

TABLE C-3. Charge sharing events affecting magnitude sensing

Time (n)	Event	Controls	Affected Nodes
7	Initial conditions		
8	Turn off precharge	P0	B00, B01
9	Turn off dummy wordline	DB00	B00, B01
10	Turn on wordline	WL	B00, B01
11	Turn on C1 to create reference	C1	B00, B01, B11
12	Turn off C1	C1	B00, B01, B11
13	Turn on I1	I1	B10, B11, SA11, SA10

Initially during sign sensing the sense amplifier is turned on and the voltage V_{SIGN} is driven onto sub-bitline B00. When the wordline is lowered there is some charge injected so

$$V_{MC}(7) = V_{SIGN} + \frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{SIGN})}{C_C} \tag{C.68}$$

The starting state for the bitlines B00 and B01 is precharge, with P0, E0 and DB00 asserted. B10 and B11 remain unchanged.

$$V_{B00}(7) = V_{BLP0} \tag{C.69}$$

$$V_{B01}(7) = V_{BLP0} \tag{C.70}$$

$$V_{B10}(7) = V_{B10}(6) \tag{C.71}$$

$$V_{B11}(7) = V_{B11}(6) \tag{C.72}$$

Turn off P0:

$$V_{B00}(8) = V_{B00}(7) + \frac{-\frac{1}{2}Q_{iP0}}{C_C + C_B} \tag{C.73}$$

$$V_{B01}(8) = V_{B01}(7) + \frac{-\frac{1}{2}Q_{iP0}}{C_C + C_B} \tag{C.74}$$

$$V_{B10}(8) = V_{B10}(7) \tag{C.75}$$

$$V_{B11}(8) = V_{B11}(7) \tag{C.76}$$

Turn off dummy wordline:

$$V_{B00}(9) = V_{B00}(8) + \frac{\left(-\frac{1}{2}\right)(Q_{iD0T})}{2C_B + C_C} \tag{C.77}$$

$$V_{B01}(9) = V_{B01}(8) + \frac{\left(-\frac{1}{2}\right)(Q_{iD0T})}{2C_B + C_C} \tag{C.78}$$

$$V_{B10}(9) = V_{B10}(8) \tag{C.79}$$

$$V_{B11}(9) = V_{B11}(8) \tag{C.80}$$

Turn on wordline

$$V_{B00}(10) = \frac{(2C_B + C_C)V_{B00}(9) + C_C V_{MC}(7) + Q_{iWL}}{2C_B + 2C_C} \tag{C.81}$$

$$V_{B01}(10) = \frac{(2C_B + C_C)V_{B01}(9) + C_C V_{MC}(7) + Q_{iWL}}{2C_B + 2C_C} \tag{C.82}$$

$$V_{B10}(10) = V_{B10}(9) \tag{C.83}$$

$$V_{B11}(10) = V_{B11}(9) \tag{C.84}$$

Turn on C1 to generate reference voltage. Only B11 and B10 are of interest from now on.

$$V_{B10}(11) = V_{B10}(10) \tag{C.85}$$

$$V_{B11}(11) = \frac{(2C_B + 2C_C)V_{B00}(10) + (C_B + C_C)V_{B11}(10) + Q_{iC1}}{3C_B + 3C_C} \tag{C.86}$$

Turn off C1 in preparation for magnitude sensing.

$$V_{B01}(12) = V_{B01}(11) \tag{C.87}$$

$$V_{B11}(12) = V_{B11}(11) + \frac{-\frac{1}{2}Q_{iC1}}{C_B + C_C} \tag{C.88}$$

Turn on I1 to share the bitlines into the sense amplifier:

$$V_{B10}(13) = \frac{(C_C + C_B)V_{B10}(12) + C_{SA}V_{SA10}(5) + Q_{i1}}{C_C + C_B + C_{SA}} \tag{C.89}$$

$$V_{B11}(13) = \frac{(C_C + C_B)V_{B11}(12) + C_{SA}V_{SA11}(5) + Q_{i1}}{C_C + C_B + C_{SA}} \tag{C.90}$$

Again it is assumed that the sense amplifier nodes are precharged and equal therefore the differential magnitude signal V_{MD} is

$$V_{MD} = V_{B10}(13) - V_{B11}(13) = \left(\frac{(C_C + C_B)}{C_C + C_B + C_{SA}}\right)(V_{B10}(12) - V_{B11}(12)) \tag{C.91}$$

Reducing each component, $V_{B10}(12)$ is

$$V_{B10}(12) = \tag{C.92}$$

$$\left(\frac{C_B}{C_B+C_C}\right)\left(\frac{V_{BLP0}}{2}+\frac{\left(-\frac{1}{2}\right)(Q_{iP0})}{2C_B+2C_C}+\frac{\left(-\frac{1}{2}\right)(Q_{iE0})}{2C_B+2C_C}\right)+\frac{Q_{iWL}+\left(-\frac{1}{2}\right)(Q_{iDoT})}{2C_B+2C_C}$$

$$+\frac{C_C V_{DATA}}{2C_B+2C_C}+\frac{C_C V_{MC}(ERROR)}{2C_B+2C_C}+\frac{V_{BLP1}}{2}+\frac{\left(-\frac{1}{2}\right)(Q_{iP1})}{2C_B+2C_C}+\frac{\left(-\frac{1}{2}\right)(Q_{iE1})}{2C_B+2C_C}$$

and $V_{B11}(12)$ is

$$V_{B11}(12) = \left(\frac{1}{3C_B+3C_C}\right)\left((2C_B+C_C)V_{BLP0}+(C_B+C_C)V_{BLP1}\right. \tag{C.93}$$

$$+C_C V_{SIGN}+\left(-\frac{1}{2}\right)(Q_{iWL}(V_{SIGN}))+\left(-\frac{1}{2}\right)Q_{iDOT}+Q_{iWL}$$

$$\left.+\left(-\frac{1}{2}\right)Q_{iP1}+\frac{\left(-\frac{1}{2}\right)(2C_B+C_C)Q_{iP0}}{C_B+C_C}+\left(-\frac{1}{2}\right)Q_{iE1}+\left(-\frac{1}{2}\right)Q_{iC1}\right)$$

And using Eq. C.57, Eq. C.58 and Eq. C.58 and assuming that $V_{BLP0} = V_{BLP1} = V_{BLP}$

$$V_{MD} = \left(\frac{1}{C_C+C_B+C_{SA}}\right)\left(\frac{1}{6}\right)\left(3C_C V_{DATA}-2C_C V_{SIGN}-C_C V_{BLP}\right. \tag{C.94}$$

$$+\frac{1}{12}Q_{iW}+Q_{iWL}(V_{SIGN})-3C_C V_{MC}(ERROR)$$

$$\left.+\frac{C_C Q_{iP}}{2C_B+2C_C}+\frac{(-4C_B-C_C)Q_{iE}}{12(C_B+C_C)}+Q_{iC}\right)$$

Splitting V_{MD} into signal and error components:

$$V_{MD}(SIGNAL) = \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{3V_{DATA} - 2V_{SIGN} - V_{BLP}}{6}\right) \quad \text{(C.95)}$$

$$V_{MD}(ERROR) = \left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{12}Q_{iW} + \frac{1}{6}Q_{iWL}(V_{SIGN}) - \frac{1}{2}C_C V_{MC}(ERROR)\right. \quad \text{(C.96)}$$
$$\left. + \left(\frac{1}{12}\right)\left(\frac{C_C}{C_B + C_C}\right)Q_{iP} + \left(\frac{1}{12}\right)\left(\frac{-4C_B - C_C}{C_B + C_C}\right)Q_{iE} + \frac{1}{6}Q_{iC}\right)$$

APPENDIX D: CHARGE INJECTION SIMULATION

The reduced noise margins of MLDRAM require that the signal presented to the sense amplifiers be free of errors introduced from unbalanced charge injection. Simulations of the Gillingham scheme revealed that charge injection effects cause a significant error in the sensed signals, depleting the signal strength by up to fifty percent. The results of a detailed charge injection analysis (Appendix C) are presented here. The goal is to understand the source of any errors in the restore process and of signal errors just prior to both sign and magnitude sensing. An analysis of charge injection effects on signal strength is presented and the extent of those effects is demonstrated through simulation. The analytical and the experimental results are considered together to identify a practical design strategy for the circuit with respect to the cell and bitline capacitances, transistor sizes and transistor types.

D.1 Charge Injection Analysis

A detailed charge injection analysis is given in Appendix C. The analysis gives predictions of the restored voltage, and the signals presented to the magnitude sense amplifiers. Table D-1 shows brief definitions for the symbols used in signal equations in the analysis and discussion. The signal equations are given as differential voltages on the bitlines that contain both the ideal voltage and the error voltage due to charge injection.

The schematic in Figure D.1 shows the Gillingham scheme with the signal and circuit node names used in the analysis.

TABLE D-1. Symbol definitions used in the analysis

Symbol	Definition
V_{SIGN}	The voltage of the sign bit: either V_{DD} or 0V.
V_{MAG}	The voltage of the magnitude bit: either V_{DD} or 0.V
V_{DATA}	The voltage of the multilevel (2-bit) data: either V_{DD}, $\frac{2}{3}V_{DD}$, $\frac{1}{3}V_{DD}$, or 0V.
V_x	The voltage on node x
$V_x(n)$	The steady-state voltage on circuit node x after timing event n
$V_x(SIGNAL)$	The ideal voltage value of parameter V_x
$V_x(ERROR)$	The difference between the ideal and the actual voltage value of parameter V_x
$Q_{im}(V_S)$	The charge injected into the channel of transistor m after the device is switched on
Q_{im}	When $V_S = V_{BLP}$, this is the charge injected into the channel of transistor m after the device is switched on
C_x	The total capacitance of circuit node x.

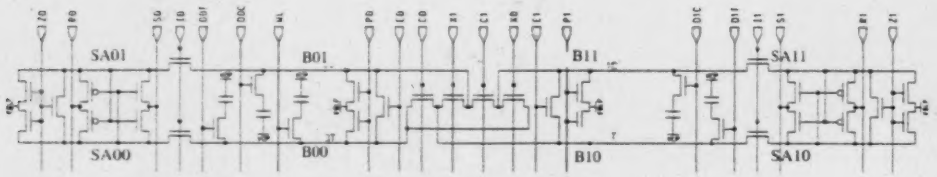


FIGURE D.1. Multilevel DRAM circuit configuration

The differential signal V_{SD} presented to the sense amplifier for sign sensing is

$$V_{SD} = V_{SD}(SIGNAL) + V_{SD}(ERROR) \tag{D.1}$$

where,

$$V_{SD}(SIGNAL) = \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{V_{DATA} - V_{BLP}}{2}\right) \quad \text{(D.2)}$$

$$V_{SD}(ERROR) = \left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{4}Q_{iW} + \frac{C_C(Q_{iP} + Q_{iE})}{4C_B + 4C_C} + \frac{C_C}{2}V_{MC}(ERROR)\right) \quad \text{(D.3)}$$

Similarly, the signal presented to the magnitude sense amplifier is

$$V_{MD} = V_{MD}(SIGNAL) + V_{MD}(ERROR) \quad \text{(D.4)}$$

where,

$$V_{MD}(SIGNAL) = \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{3V_{DATA} - 2V_{SIGN} - V_{BLP}}{6}\right) \quad \text{(D.5)}$$

$$V_{MD}(ERROR) = \left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{12}Q_{iW} + \frac{1}{6}Q_{iWL}(V_{SIGN}) - \frac{1}{2}C_C V_{MC}(ERROR)\right. \quad \text{(D.6)}$$
$$\left. + \left(\frac{1}{12}\right)\left(\frac{C_C}{C_B + C_C}\right)Q_{iP} + \left(\frac{1}{12}\right)\left(\frac{-4C_B - C_C}{C_B + C_C}\right)Q_{iE} + \frac{1}{6}Q_{iC}\right)$$

These equations give some indication of the effect of the charge injected via the various switches into the sensed signals. The equations are verified through simulation in the next section.

D.2 Simulation

D.2.1. Overview

The object of the simulation is to verify the contributions of the various terms in the sensing input voltages V_{SD} and V_{MD}. To do this a full simulation of the Gillingham scheme was set up in the 0.35 µm process. The values of Q_{im} cannot be measured directly in simulation, but the charge injected by each switch can be inferred by noting the voltage change to a known capacitance caused by the switch being toggled. In this way each of the

injection values is measured from simulation and the final signal and error voltages are predicted. The predicted signals are then compared with the final V_{SD} and V_{MD} voltages to verify the equations.

D.2.2. Setup

The lengths and widths of the switch transistors are shown in Table D-2. The supply voltages are given in Table D-3.

TABLE D-2. Transistor sizes and gate voltages used for switching

Transistor	Type	Length (um)	Width (um)	V_{ON} (V)
I	nmos	0.350	2.100	5.0
P	nmos	0.350	1.750	3.3
E	nmos	0.350	1.750	5.0
C	nmos	0.350	1.750	3.3
X	nmos	0.350	1.750	5.0
WL	pmos	0.350	0.900	-1.75
Cell Cap	pmos	1.465	4.550	N/A

TABLE D-3. Supply Voltages

Voltage	Value (V)
V_{DD}	3.3
V_{SS}	0
V_{BLP}	1.65
V_{CP}	4.5

The timing from Gillingham's paper was used but the delays between edges were increased so that the bitlines have sufficient time to settle at their steady state voltages between timing events. Through simulation, a delay of 10 ns between edges was found to be sufficient. For charge sharing operations, where large voltages are involved (e.g. asserting E0 for restore), a delay of 20 ns was found to be sufficient and a delay of 40 ns was allowed for sensing. Measurements indicate that these delays allow equalization and settling to within 1 μV.

D.2.3. Measuring Important Capacitance Ratios

To analyze the simulation results the ratios of C_C to C_B and C_B to C_{SA} are needed.

For this process

$$C_{OX} = 4.8\frac{fF}{um^2} \tag{D.7}$$

So the cell capacitance is found from Eq. C.1 to be

$$C_C = WLC_{OX} = 32fF \tag{D.8}$$

The error component V_{SD}(ERROR) of the sign signal has only one data-dependent term. That term comes from the non-ideal restore value. In the beginning of the simulation, the cell contains only ideal values so the error in V_{SD} is constant regardless of the data. Removing the data-dependent term from Eq. D.3 gives:

$$V_{SD}(ERROR) = -\left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{4}Q_{iW} + \frac{C_C(Q_{iP} + Q_{iE})}{4C_B + 4C_C}\right) \tag{7.1}$$

V_{SD}(ERROR) can be cancelled out mathematically by subtracting two different V_{SD} values. When the signal (V_{SD}) of Data = 11 is subtracted from that of Data = 00, the difference contains no error components and is equal to:

$$V_{SD}(11) - V_{SD}(00) = \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{V_{DD} - V_{BLP}}{2} - \frac{V_{SS} - V_{BLP}}{2}\right) \tag{D.9}$$

$$= \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{V_{DD} - V_{SS}}{2}\right) \tag{D.10}$$

From the definition of V_{DATA} and V_{BLP} the following can be concluded for Data = 11:

$$V_{SD}(SIGNAL, 11) = \frac{V_{SD}(11) - V_{SD}(00)}{2} = \frac{1}{2}\left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{V_{DD} - V_{SS}}{4}\right) \tag{D.11}$$

Similarly, for the other three Data values:

$$V_{SD}(SIGNAL,00) = -\frac{V_{SD}(11)-V_{SD}(00)}{2} = -\frac{1}{2}\left(\frac{C_C}{C_C+C_B+C_{SA}}\right)\left(\frac{V_{DD}-V_{SS}}{4}\right) \quad (D.12)$$

$$V_{SD}(SIGNAL,10) = \frac{V_{SD}(10)-V_{SD}(01)}{2} = \frac{1}{6}\left(\frac{C_C}{C_C+C_B+C_{SA}}\right)\left(\frac{V_{DD}-V_{SS}}{4}\right) \quad (D.13)$$

$$V_{SD}(SIGNAL,01) = -\frac{V_{SD}(10)-V_{SD}(01)}{2} = -\frac{1}{6}\left(\frac{C_C}{C_C+C_B+C_{SA}}\right)\left(\frac{V_{DD}-V_{SS}}{4}\right) \quad (D.14)$$

These signal values can all be measured from simulation without knowing the actual values of C_C or C_B. From them the capacitance ratio can be calculated. The simulated signal values for the four possible V_{SD} values are given in Table D-4. The ideal signals are calculated from Eqs. D.6 to D.9 and shown in Table D-5. V_{MD} is also readily available from these results since V_{MD} has the same magnitude as V_{SD}(SIGNAL,01).

TABLE D-4. Raw measurements from simulation

	$V_{SD}(00)$	$V_{SD}(01)$	$V_{SD}(10)$	$V_{SD}(01)$
Value (mV)	-151.68	-53.14	45.70	145.18

TABLE D-5. Ideal signal values

	Data = 00	Data = 01	Data = 10	Data = 11
V_{SD} (SIGNAL)	-148.43	-49.42	49.42	148.43
V_{MD} (SIGNAL)	-49.42	49.42	-49.42	49.42

From Table D-5 the capacitance ratio $\frac{C_C}{C_C+C_B}$ can be determined by noting the following:

$$\left(\frac{C_C}{C_C+C_B+C_{SA}}\right) = \left(\frac{C_C+C_B}{C_C+C_B+C_{SA}}\right)\left(\frac{C_C}{C_C+C_B}\right) = K_1K_0 \quad (D.15)$$

where K_1 is defined is a scaling factor due to charge-sharing the sub-bitlines with the sense amplifiers, and K_0 is the scaling factor due to charge sharing the cell with the sub-bitline. The definitions of these two factors are:

$$K_1 = \frac{C_C + C_B}{C_C + C_B + C_{SA}} \tag{D.16}$$

$$K_0 = \frac{C_C}{C_C + C_B} \tag{D.17}$$

K_0 is then calculated by:

$$K_0 = \frac{1}{K_1} V_{SD}(SIGNAL, 11)(2)\left(\frac{4}{V_{DD} - V_{SS}}\right) \tag{D.18}$$

A similar calculation can be performed for each V_{SD}(SIGNAL, x) for x = 00, 01 and 10. K_1 is calculated from simulation measurements by using the differential voltage on the bitlines before and after the isolation device is turned on:

$$K_1 = \frac{(V_{0T}(6) - V_{0C}(6))}{(V_{0T}(5) - V_{0C}(5))} \tag{D.19}$$

The simulation results of these measurements of and the calculations are shown in Table D-6.

TABLE D-6. Measurements to determine K_0 and K_1

Data	00	01	10	11
$V_{OT}(5)$-$V_{OC}(5)$, (mV)	-180.00	-63.05	54.20	172.05
$V_{OT}(6)$-$V_{OC}(6)$, (mV)	-151.68	-53.14	45.70	145.18
K_1 (dimensionless)	0.8427	0.8428	0.8432	0.8438
K_0 (dimensionless)	0.2135	0.2132	0.2132	0.2135

The average value for K_1 to 0.1% accuracy is

$$K_1 = 0.843 \tag{D.20}$$

The average value for K_0 to 0.1% accuracy is

$$K_0 = 0.213 \tag{D.21}$$

D.2.4. Multilevel Restore

The equations of interest for multilevel restore are:

$$V_{MC}(IDEAL) = V_{DATA} = \frac{2V_{SIGN} + V_{MAG}}{3} \tag{D.22}$$

$$V_{MC}(ERROR) = \frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C} + \frac{\left(-\frac{1}{2}\right)Q_{i10}(V_{SIGN}) + \left(-\frac{1}{2}\right)Q_{i11}(V_{MAG}) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} \tag{D.23}$$

All of the measurements needed to calculate the final memory cell voltage after restore (V_{MC}) are given in Table D-7. The data was acquired by measuring the bitline voltages before and after a switch is turned off. The difference in these two voltages is caused by charge injection. One exception to this is when E1 is turned on to charge share. Here the injection voltage was calculated by measuring $V_{B00}(-3)$ and $V_{B01}(-3)$ (from Appendix B) before and after the switch is turned on. Eq. AC.18 was used to calculate the voltage change caused by injection from E1.

TABLE D-7. Raw measurements of charge injection taken from simulation

Event (n)	Voltage	Time (ns)	Value (mV) (Data = 00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
-5	$\frac{\left(-\frac{1}{2}\right)Q_{i11}(V_{MAG})}{C_B + C_C}$	290	-59.0	-22.5	-59.0	-22.5
-3	$\frac{\left(-\frac{1}{2}\right)Q_{i10}(V_{SIGN})}{2C_B + 2C_C}$	340	-28.6	-28.6	-2.54	-2.54

TABLE D-7. Raw measurements of charge injection taken from simulation

Event (n)	Voltage	Time (ns)	Value (mV) (Data = 00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
-2	$\frac{Q_{iE1}(V_{DATA})}{3C_B + 3C_C}$	350	29.6	39.8	53.7	8.56
-1	$\frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C}$	370	43.2	68.9	89.3	104.4

The final cell voltage error is calculated from Table D-7 and Eq. D.23. Each term of Eq. D.23 was calculated separately and appears in Table D-8. Each is given a partial error name of the form V_{MC}(ERROR, M), where M represents the name of the transistor name that caused the error term. The second last row of Table D-8 is the sum of the error terms and the last row is the measured signal errors.

TABLE D-8. Final results from simulation of multilevel restore

Partial Error Name	Term of Eq. 25	Value (mV) (Data = 00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
Measured V_{MC}(ERROR, I1)	$\frac{\left(-\frac{1}{2}\right)Q_{iI1}(V_{MAG})}{3C_B + 3C_C}$	-19.7	-7.5	-19.7	-7.50
Measured V_{MC}(ERROR, I0)	$\frac{\left(-\frac{1}{2}\right)Q_{iI0}(V_{SIGN})}{3C_B + 3C_C}$	-19.1	-19.1	-3.81	-3.81
Measured V_{MC}(ERROR, E1)	$\frac{Q_{iE1}(V_{DATA})}{3C_B + 3C_C}$	29.6	39.8	53.7	8.56
Measured V_{MC}(ERROR, WL)	$\frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C}$	43.2	68.9	89.3	104.4
Calculated V_{MC}(ERROR)		34.0	82.1	119.5	101.7
Measured V_{MC}(ERROR)		31.3mV	79.5mV	116mV	98mV

One peculiar result in Table D-8 is the error term V_{MC}(ERROR, E1). In theory this error should be linear with respect to the data voltage and it should not be any more significantly than the others (except V_{MC}(ERROR, WL)). There are two data points that are not consistent: the values for Q_{iE1} taken at n = -2 for Data = 01 and Data = 10 do not fit

near a line between Data=00 ad Data=11. One expects that when Data = 00, the charge injection for the NMOS device E1 would be at its highest, while when Data = 11 that charge injection would be lowest. The points between should fall within these extremes but they do not. The discrepancy may be attributed to two factors. First, there is an assumption that the channel capacitance C_{GX0} does not participate in charge sharing when in fact it does. Second, a component of C_B is a voltage-dependent capacitance due to the source of each deactivated cell access transistor connected to the sub-bitlines. To quantify the contribution of these higher order errors, each will be considered independently.

First, the assumption that the channel capacitance C_{GX0} is too small to have a significant impact on charge sharing operations is justified when one considers that, for this simulation, C_B = 105 fF, C_C = 32 fF and C_{GX0} = 2.9 fF so

$$C_{GX0} \ll 3C_B + 3C_C \tag{D.24}$$

However, C_{GX0} does participate in charge sharing and the contribution to the error can be determined. Consider the case when Data=01 and ignore all charge injection errors. At n=-3 the sub-bitlines 0T and1C are charged to 0 V, the switch X0 is on and the sub-bitline 1T is charged to 3.3 V. The total charge in the system is:

$$Q_{initial} = (2C_B + 2C_C + C_{GX0})(3.3V) + (C_B + C_C)(0V) \tag{D.25}$$

The switch E1 is asserted and the total capacitance in the system is

$$C_{final} = 3C_B + 3C_C + C_{GX0} \tag{D.26}$$

This gives a final voltage of

$$V_{final} = V_{0T}(-2) = \frac{Q_{final}}{C_{final}} = \frac{(2C_B + 2C_C + C_{GX0})(V_{SIGN}) + (C_B + C_C)(V_{MAG})}{3C_B + 3C_C + C_{GX0}} \tag{D.27}$$

$$V_{final} = \frac{(2C_C + K_0 C_{GX0})(3.3V) + (C_C)(0V)}{3C_C + K_0 C_{GX0}} = 2.207V \tag{D.28}$$

The ideal value is 2.2 V. Thus there is about 7 mV between what is expected and what is simulated. The results of this calculation for all data values are shown in Table D-9. There the error associated with this phenomenon is denoted as $V_{MC}(ERROR, C_{GX0})$.

TABLE D-9. Errors from capacitance imbalance due to C_G of X0

Data	**00**	**01**	**10**	**11**
V_{final}	0.0000	1.0930	2.2070V	3.3000
$V_{expected}$	0.0000	1.1000	2.2000	3.3000
$V_{MC}(ERROR, C_{GX0}) = V_{final} - V_{expected}$	0.0mV	-7.0mV	+7.0mV	0.0mV

The rest of the discrepancy can be attributed to the non-linear capacitance of C_B. Distributed along each sub-bitline are 32 cell access transistors operating in the cutoff region. To the sub-bitline these appear as reverse biased p-n junctions and thus act as voltage-dependent capacitors. As the voltage difference between the unused wordline (V_{WL}) and the sub-bitline (V_B) changes, the width of the depletion region of each p-n junction changes. As $V_{WL} - V_B$ increases, the width of the depletion increases thus the capacitance decreases. When considering the cases where Data = 01 and Data = 10 there are four possible values of C_B involved. For example, when restoring Data = 10, the two sub-bitlines carrying the sign have maximum capacitance ($V_{WL} - V_B = 0$) and the sub-bitline carrying the magnitude bit has minimal capacitance (V_{WL} $V_B = 3.3V$). Furthermore, the final voltage from charge sharing these together at n = -2 is complicated by the fact that the total final capacitance is a function of the voltage as well. To fully evaluate the final voltage requires incorporating the nonlinear capacitance of each cell access transistor: The nonlinear portion of the sub-bitline capacitance can be expressed as

$$C \propto (V_{GS} + K)^{-\frac{1}{3}} \tag{D.29}$$

where K depends on the process. This analysis is not performed here. Instead, a new error term $V_{MC}(ERROR, C_B)$ can be defined to denote the error associated with the nonlinearity of C_B.

An important aspect of the behavior of V_{MC}(ERROR, C_{GX0}) and V_{MC}(ERROR, C_B) is that there is no effect on the final voltage when Data = 00 and Data = 01 because the initial and final voltages are the same. Thus, no charge sharing occurs. For V_{MC}(ERROR, C_B) there is no change in C_B between n=-3 and n=-2 because the bitline voltage does not change. In other words, these effects are not seen when $V_{SIGN} = V_{MAG}$. With this observation, the real value of the charge injection can be calculated by linear interpolation:

$$V_{MC}(ERROR, E1) = \frac{Q_{iE1}(V_{DATA})}{3C_B + 3C_C} = -6.3758 V_{DATA} + 29.6 \qquad \text{(D.30)}$$

The components that make up the unexpected error results for V_{MC}(ERROR, E1) of Table D-8 are decomposed in Table D-10. The contribution from the nonlinear C_B effect is calculated for each data value and shown in Table D-10. About +30mV of error is introduced into the restored voltage due to the nonlinear behavior of C_B.

TABLE D-10. Decomposition of the measured error at n = -2

Voltage	Value (mV) (Data = 00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
Measured V_{MC}(ERROR, E1)	29.6	39.8	53.7	8.56
Interpolated V_{MC}(ERROR, E1)	29.6	22.6	15.6	8.56
Contribution from V_{MC}(ERROR, C_{GX0})	0	-7.0	7.0	0
Contribution from V_{MC}(ERROR, C_B)	0	24.2	31.1	0

Based on these findings, Eq. D.3 can be modified to include the two nonlinear effects. The modified form of Eq. D.3 is shown below. Eq. D.31 contains all of the separate errors of Tables D-9 and D-10.

$$\begin{aligned} V_{MC}(ERROR) = {} & V_{MC}(ERROR, I0) + V_{MC}(ERROR, I1) + V_{MC}(ERROR, E1) \\ & + V_{MC}(ERROR, WL) + V_{MC}(ERROR, I0) \\ & + V_{MC}(ERROR, C_{CX0}) + V_{MC}(ERROR, C_B) \end{aligned} \qquad \text{(D.31)}$$

D.2.5. Sign Sensing

The equations of interest for sign sensing are repeated here:

$$V_{SD}(SIGNAL) = \left(\frac{C_C}{C_C + C_B + C_{SA}}\right)\left(\frac{V_{DATA} - V_{BLP}}{2}\right) \tag{D.32}$$

$$V_{SD}(ERROR) = -\left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{4}Q_{iW} + \frac{C_C(Q_{iP} + Q_{iE})}{4C_B + 4C_C} + \frac{C_C}{2}V_{MC}(ERROR)\right) \tag{D.33}$$

The final restore value from Table D-9 and measurements from simulation are given in Table D-11 and are used to calculate the values of Eq. D.33.

TABLE D-11. Measurements taken from sign sensing simulation

Event (n)	Voltage	Time (ns)	Value (Data=00)	Value (Data=01)	Value (Data=10)	Value (Data=11)
-1	$V_{MC}(ERROR)$	380	31.3mV	79.5mV	116mV	98mV
1	$\frac{\left(-\frac{1}{2}\right)(Q_{iP0} + Q_{iE0})}{C_C + C_B}$	440	-37.2mV	-37.2mV	-37.2mV	-37.2mV
2	$V_{0T}(3) - V_{0T}(2)$	30	-384.93mV	-150.38mV	84.09mV	318.83mV

The error terms of Eq. D.33 are calculated as follows:

$$\frac{\frac{C_C}{2}V_{MC}(ERROR)}{C_C + C_B + C_{SA}} = (V_{MC}(ERROR))\left(\frac{K_1 K_0}{2}\right) \tag{D.34}$$

$$\frac{\left(\frac{C_C(Q_{iP} + Q_{iE})}{4C_B + 4C_C}\right)}{C_C + C_B + C_{SA}} = -\left(\frac{\left(-\frac{1}{2}\right)(Q_{iP0} + Q_{iE0})}{C_C + C_B}\right)\left(\frac{K_1 K_0}{2}\right) \tag{D.35}$$

$$\frac{\frac{1}{4}Q_{iW}}{C_C + C_B + C_{SA}} = \frac{K_0}{4}(V_{0T}(3) - V_{0T}(2) - K_1(V_{DATA} - V_{BLP})) \tag{D.36}$$

The calculations are collected in Table D-12.

TABLE D-12. Calculated errors in sign sensing

Error	Term of Eq. 126	Value (mV) (Data = 00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
V_{SD}(ERROR, V_{MC})	$\frac{\frac{C_C}{2}V_{MC}(ERROR)}{C_C+C_B+C_{SA}}$	2.81	7.14	10.4	8.80
Measured V_{SD}(ERROR, P, E)	$\frac{\left(\frac{C_C(Q_{iP}+Q_{iE})}{4C_B+4C_C}\right)}{C_C+C_B+C_{SA}}$	3.34	3.34	3.34	3.34
Measured V_{SD}(ERROR, W)	$\frac{\frac{1}{4}Q_{iW}}{C_C+C_B+C_{SA}}$	-7.06	-7.00	-6.97	-6.87
Total V_{SD}(ERROR)		-0.91	3.48	6.8	5.27
Measured V_{SD}(ERROR)		-0.43	3.41	6.72	5.59

Just to check that the totals are correct, the total error can be measured by comparing the final signal V_{SD} to the ideal value from Table D-5. From the last two rows of Table D-12 the discrepancy between the calculations and the measurements is less than 0.5 mV. The components of the signal V_{SD} are summarized below in Table D-13.

TABLE D-13. Final V_{SD} components just before sign sensing.

	Value (mV) (Data = 00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
Measured V_{SD}	-148.88	-46.01	56.14	154.02
V_{SD}(SIGNAL) from Table D-	-148.43	-49.42	49.42	148.43
V_{SD}(ERROR)	-0.43	3.41	6.72	5.59

D.2.6. Magnitude Sensing

The equations of interest for sign sensing, Eqs. 95 and 96, are repeated below. The analysis follows the same procedure as for the previous two sections. Table D-14 shows all of the raw measurements taken from simulation that are useful in determining the error components. Eqs. D.39 to D.43 are used to calculate the required error terms from the data

in Table D-14 and the results of the calculations, as well as the measurement of V_{MD}, are given in Table D-15.

$$V_{MD}(SIGNAL) = K_0 K_1 \left(\frac{3V_{DATA} - 2V_{SIGN} - V_{BLP}}{6} \right) \tag{D.37}$$

$$V_{MD}(ERROR) = \left(\frac{1}{C_C + C_B + C_{SA}} \right) \left(\frac{1}{12} Q_{iW} + \frac{1}{6} Q_{iWL}(V_{SIGN}) - \frac{1}{2} C_C V_{MC}(ERROR) \right. \tag{D.38}$$
$$\left. + \left(\frac{1}{12} \right) \left(\frac{C_C}{C_B + C_C} \right) Q_{iP} + \left(\frac{1}{12} \right) \left(\frac{-4C_B - C_C}{C_B + C_C} \right) Q_{iE} + \frac{1}{6} Q_{iC} \right)$$

TABLE D-14. Measurements taken from magnitude sensing

Voltage	Appendix	Value (Data = 00)	Value (Data = 01)	Value (Data = 10)	Value (Data = 11)
$\frac{\frac{1}{4}Q_{iW}}{C_C + C_B + C_{SA}}$		-7.06	-7.00	-6.97	-6.87
$V_{MC}(7) - V_{MC}(6)$		43.01mV	43.01mV	105.51mV	105.47mV
$V_{MC}(ERROR)$		31.3mV	79.5mV	116mV	98mV
$V_{0T}(8) - V_{0T}(7)$		-10.48mV	-10.48mV	-10.48mV	-10.48mV
$\frac{\left(-\frac{1}{2}\right)Q_{iE0}}{C_C + C_B}$		-29.62mV	-29.62mV	-27.30mV	-27.30mV
$V_{0T}(12) - V_{0T}(11)$		-12.89mV	-12.89mV	-11.37mV	-11.37mV

$$\frac{\frac{1}{12}Q_{iW}}{C_C + C_B + C_{SA}} = \frac{K_1}{6}\left(V_{0T}(10) - \left(1 - \frac{K_0}{2}\right) V_{0T}(9) - \frac{K_0}{2} V_{MC}(7) \right) \tag{D.39}$$

$$\frac{\frac{1}{6}Q_{iWL}(V_{SIGN})}{C_C + C_B + C_{SA}} = -\frac{K_1 K_0}{3}(V_{MC}(7) - V_{MC}(6)) \tag{D.40}$$

$$\left(\frac{1}{12}\right)\left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{C_C}{C_B + C_C}\right) Q_{iP} = -\frac{K_0 K_1}{6}((V_{0T}(8) - V_{0T}(7))) = \left(\frac{K_0 K_1}{12}\right)\left(\frac{Q_{iP}}{C_B + C_C}\right) \tag{D.41}$$

$$\left(\frac{1}{12}\right)\left(\frac{1}{C_C+C_B+C_{SA}}\right)\left(\frac{-4C_B-C_C}{C_B+C_C}\right)Q_{iE} = \left(-\frac{K_1}{6}\right)(3K_0-4)\left(\frac{\left(-\frac{1}{2}\right)Q_{iE0}}{C_C+C_B}\right) = \left(-\frac{K_1}{12}\right)\left(\frac{3K_0-4}{C_B+C_C}\right)Q_{iE} \quad \text{(D.42)}$$

$$\frac{\frac{1}{6}Q_{iC}}{C_C+C_B+C_{SA}} = -\frac{2}{3}(V_{0T}(12)-V_{0T}(11)) \quad \text{(D.43)}$$

TABLE D-15. Calculated errors for magnitude sensing

Error	Term of Eq. 126	Value (mV) (Data=00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
V_{MD}(ERROR, V_W)	$\frac{\frac{1}{12}Q_{iW}}{C_C+C_B+C_{SA}}$	-2.33	-2.33	-2.33	-2.33
V_{MD}(ERROR,W,V_{SIGN})	$\frac{\frac{1}{6}Q_{iWL}(V_{SIGN})}{C_C+C_B+C_{SA}}$	-2.57	-2.57	-6.31	-6.31
V_{MD}(ERROR, P)	$\left(\frac{K_0K_1}{12}\right)\left(\frac{Q_{iP}}{C_B+C_C}\right)$	0.31	0.31	0.31	0.31
V_{MD}(ERROR, E)	$\left(-\frac{K_1}{12}\right)\left(\frac{3K_0-4}{C_B+C_C}\right)Q_{iE}$	-13.98	-13.98	-12.89	-12.89
V_{MD}(ERROR, V_{MC})	$\frac{1}{2}K_0K_1V_{MC}(ERROR)$	2.81	7.14	10.41	8.78
V_{MD}(ERROR, C)	$\frac{\frac{1}{6}Q_{iC}}{C_C+C_B+C_{SA}}$	8.59	8.59	7.58	7.58
Total V_{MD}(ERROR)		-7.17	-2.84	-3.23	-4.86

Again the errors agree. The affects of approximations are apparent in V_{MD}(ERROR,C) since this should be constant across all data values. The last row of Table D-15 can be compared to Table D-16. The differences might be attributed to the approximations that

the channel does not participate in charge sharing and that the bitline capacitance is non-linear.

TABLE D-16. Final V_{MD} components just before sensing

Error	Value (mV) (Data=00)	Value (mV) (Data = 01)	Value (mV) (Data = 10)	Value (mV) (Data = 11)
Measured V_{SD}	-57.20	45.64	-54.92	43.08
V_{MD}(SIGNAL) from Table D-	-49.42	49.42	-49.42	49.42
V_{MD}(ERROR)	-7.78	-3.78	-5.50	-6.34

D.3 Design Method to Eliminate Charge Injection Effects.

The results of the simulation verify that the equations work in practice. The error terms of the equations depend on the charge injection which, in turn, depends on the transistor size and type (NMOS or PMOS). To eliminate charge injection error, the circuit should be designed such that

$$V_{MC}(ERROR) = \frac{\left(-\frac{1}{2}\right)Q_{iWL}(V_{DATA})}{C_C} + \frac{\left(-\frac{1}{2}\right)Q_{iI0}(V_{SIGN}) + \left(-\frac{1}{2}\right)Q_{iI1}(V_{MAG}) + Q_{iE1}(V_{DATA})}{3C_B + 3C_C} = 0 \quad \text{(D.44)}$$

$$V_{SD}(ERROR) = -\left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{4}Q_{iW} + \frac{C_C(Q_{iP} + Q_{iE})}{4C_B + 4C_C} + \frac{C_C}{2}V_{MC}(ERROR)\right) = 0 \quad \text{(D.45)}$$

$$V_{MD}(ERROR) = \left(\frac{1}{C_C + C_B + C_{SA}}\right)\left(\frac{1}{12}Q_{iW} + \frac{1}{6}Q_{iWL}(V_{SIGN}) - \frac{1}{2}C_C V_{MC}(ERROR)\right. \quad \text{(D.46)}$$
$$\left. + \left(\frac{1}{12}\right)\left(\frac{C_C}{C_B + C_C}\right)Q_{iP} + \left(\frac{1}{12}\right)\left(\frac{-4C_B - C_C}{C_B + C_C}\right)Q_{iE} + \frac{1}{6}Q_{iC}\right) = 0$$

One possible iterative method for achieving these conditions is as follows:

(1) Set up the simulation according to the correct bitline, cell and sense amplifier specifications. (2) Size the transistors for C, X, E, P, I to be equal and of the same type. (3) Run a simulation to get K_0 and K_1. (4) Run simple simulations that toggle each switch on and off while the bitlines are at V_{BLP}. (5) Use these results to calculate each $Q_i(V_S)$ that

appears in Eqs. D.44, D.45 and D.46. (6) Calculate the coefficients of each term of Eqs. D.44, D.45 and D.46. (7) Make a spreadsheet of the results of (5) and (6) and use it to calculate the error. (8) Add weighting factors to the spreadsheet to allow manipulation of the terms. (9) Use the weighting factors to adjust the effect of each term on the errors of Eqs. D.44, D.45 and D.46. A negative weight means the transistor type should be changed.

After the errors on Eqs. D.45 and D.46. have been made as small as desired, use the final weighting factors to resize and re-type the transistors by multiplying the device width by the weighting factor.

D.4 Other Sources of Signal Error

The multilevel sense and restore scheme was re-simulated with cross-coupling capacitors in place to model metal-to-metal capacitance. At time n=2 there is an unexpected result. When the sign and magnitude bits are equal, there is a voltage bump on B10 that results from reversing the polarity of B11. For Data=00 the restored value is slightly lower than expected and of there is a 11 restore value is slightly higher than expected. This is not troublesome as it essentially enhances the signal in both cases.

When the sign bit is not the same as the magnitude bit there is another problem that becomes apparent. When E1 is activated to short out B10 and B11, the cross-coupling component of C_B (that between B10 and B11) is short-circuited through E1 and so it does not participate in charge sharing. The net effect is that there is a disproportionate weighting of the sign bit. The sign bit is weighted slightly more than it should be (i.e. exactly twice the magnitude bit). The result is that the restored value favors the sign bit. For Data = 01 the restore is a bit lower than expected and for Data = 10 the restore value is a bit higher than expected. These effects cause significant error in the restore value and so could cause errors in subsequent sensing.